



KERJA PRAKTIK - KI141330

Pembuatan SIBeATI (Sistem Informasi Beasiswa Alumni Teknik Informatika ITS)

Teknik Informatika ITS

**Jl. Teknik Kimia-Gedung Teknik Informatika Kampus ITS
Surabaya Jalan Raya ITS, Sukolilo, Surabaya 60111**

Periode: 06 Juli 2020 - 28 Agustus 2020

Oleh:

Nur Muhammad Husnul Habib Yahya

05111740000094

Carlo Win Marshal

05111740000098

Pembimbing Jurusan

Ary Mazharuddin S., S.Kom., M.Comp.Sc.

Pembimbing Lapangan

Farid Redrikana

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020



KERJA PRAKTIK - KI141330

**Pembuatan SIBeATI (Sistem Informasi Beasiswa Alumni
Teknik Informatika ITS)**

Teknik Informatika ITS

**Jl. Teknik Kimia-Gedung Teknik Informatika Kampus ITS
Surabaya Jalan Raya ITS, Sukolilo, Surabaya 60111**

Periode: 06 Juli 2020 - 28 Agustus 2020

Oleh:

Nur Muhammad Husnul Habib Yahya 05111740000094

Carlo Win Marshal 05111740000098

Pembimbing Jurusan

Ary Mazharuddin S., S.Kom., M.Comp.Sc.

Pembimbing Lapangan

Farid Redrikana

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KERJA PRAKTIK

Pembuatan SIBeATI (Sistem Informasi Beasiswa Alumni Teknik
Informatika ITS)

Oleh :

Nur Muhammad Husnul Habib Yahya 05111740000094

Carlo Win Marshal 05111740000098

Disetujui oleh Pembimbing Kerja Praktik :

1. Ary Mazharuddin S., S.Kom., M.Comp.Sc.
NIP. 19810620 100501 1 003



(Pembimbing Jurusan)

2. Farid Redrikana
Alumni Teknik Informatika ITS



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Pembuatan SIBeATI (Sistem Informasi Beasiswa Alumni Teknik Informatika ITS)

Nama Mahasiswa : Nur Muhammad Husnul Habib Yahya
NRP : 05111740000094
Nama Mahasiswa : Carlo Win Marshal
NRP : 05111740000098
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Jurusan : Ary Mazharuddin S., S.Kom., M.Comp.Sc.

ABSTRAK

SIBeATI adalah sistem informasi yang digunakan untuk mengelola donasi beasiswa dari para alumni Teknik Informatika ITS dan penyalurannya kepada para mahasiswa Teknik Informatika ITS. Sistem dibuat dengan tujuan untuk menciptakan sistem informasi terintegrasi yang dapat meningkatkan kualitas tata kelola beasiswa alumni Teknik Informatika ITS.

Sistem informasi merupakan sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, dimana sistem tersebut merupakan kombinasi dari orang-orang yang terlibat, teknologi informasi, dan prosedur-prosedur yang terorganisasi satu sama lain.

Dengan melihat adanya berbagai kebutuhan ini, pembuatan sistem informasi yang mencakup kebutuhan-kebutuhan sistem informasi harus mengikuti teknologi dan aturan-aturan tertentu.

Kata kunci:

SIBeATI, Kebutuhan Sistem Informasi

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat limpahan rahmat dan lindungan-Nya penulis dapat melaksanakan salah satu kewajiban sebagai mahasiswa Teknik Informatika ITS yaitu Kerja Praktik (KP).

Penulis menyadari masih terdapat banyak kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini, namun kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Penulis mengharapkan kritik dan saran yang membangun untuk kesempurnaan penulisan buku laporan ini.

Melalui laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada kepada pihak-pihak yang telah membantu dalam pelaksanaan kerja praktik hingga penyusunan laporan Kerja praktik baik secara langsung maupun tidak langsung. Pihak-pihak tersebut antara lain adalah:

1. Kedua orang tua dan keluarga yang senantiasa mendukung, mendoakan, serta memotivasi penulis dalam menyelesaikan Tugas akhir ini.
2. Bapak Ary Mazharuddin S., S.Kom., M.Comp.Sc. selaku dosen pembimbing kerja praktik yang telah membimbing penulis selama kerja praktik berlangsung serta selaku koordinator Kerja Praktik.
3. Bapak Farid Redrikana selaku pembimbing lapangan selama kerja praktik yang telah memberikan bimbingan serta ilmunya kepada penulis.

Surabaya, September 2020

Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK.....	VI
KATA PENGANTAR	VII
DAFTAR ISI	IX
DAFTAR GAMBAR	XIII
DAFTAR TABEL	XVI
BAB I PENDAHULUAN.....	1
1.1. LATAR BELAKANG	1
1.2. TUJUAN	2
1.3. MANFAAT	2
1.4. RUMUSAN PERMASALAHAN	3
1.5. LOKASI DAN WAKTU KERJA PRAKTIK	3
1.6. METODOLOGI KERJA PRAKTIK	3
1.7. SISTEMATIKA LAPORAN	6
BAB II PROFIL INSTANSI	9
2.1. PROFIL INSTANSI	9
BAB III TINJAUAN PUSTAKA	11
3.1. SISTEM INFORMASI	11
3.2. PHP	11
3.3. MYSQL	11
BAB IV ANALISIS DAN DESAIN	15
4.2. DIAGRAM KASUS PENGGUNAAN	16
4.3. SPESIFIKASI KASUS PENGGUNAAN	18
4.3.1. Melakukan Pencatatan Transaksi Donasi	18
4.3.2. Melakukan Pencatatan Penyaluran Beasiswa	19

4.3.3.	<i>Mengupload Dokumen Mutasi Rekening</i>	20
4.3.4.	<i>Mengedit Data Transaksi Yang Tercatat</i>	21
4.3.5.	<i>Membuat Perencanaan Penyaluran Beasiswa</i>	22
4.3.6.	<i>Menghapus Data Transaksi Yang Tercatat</i>	23
4.3.7.	<i>Melihat Perencanaan Penyaluran Beasiswa</i>	24
4.3.8.	<i>Memecah Transaksi Menjadi Banyak.....</i>	25
4.3.9.	<i>Melihat Laporan Kegiatan</i>	26
4.3.10.	<i>Melihat Laporan Keuangan</i>	27
4.3.11.	<i>Melihat Neraca Keuangan.....</i>	28
4.3.12.	<i>Melakukan Transfer Antar Akun.....</i>	29
4.3.13.	<i>Membuat Surat Perintah Penyaluran Beasiswa</i>	30
4.4.	DIAGRAM AKTIVITAS.....	32
4.4.1.	<i>Melakukan pencatatan transaksi donasi</i>	32
4.4.2.	<i>Melakukan pencatatan penyaluran beasiswa.....</i>	33
4.4.3.	<i>Mengupload dokumen berisi mutasi rekening</i>	34
4.4.4.	<i>Mengedit data transaksi.....</i>	35
4.4.5.	<i>Membuat perencanaan penyaluran beasiswa</i>	35
4.4.6.	<i>Menghapus data transaksi.....</i>	36
4.4.7.	<i>Melihat perencanaan penyaluran</i>	37
4.4.8.	<i>Memecah transaksi menjadi beberapa</i>	38
4.4.9.	<i>Melihat laporan kegiatan</i>	39
4.4.10.	<i>Melihat laporan keuangan</i>	39
4.4.11.	<i>Melihat neraca keuangan.....</i>	40
4.4.12.	<i>Melakukan transfer antar akun.....</i>	41
4.4.13.	<i>Membuat surat perintah penyaluran beasiswa</i>	42
BAB V IMPLEMENTASI SISTEM		43
5.	IMPLEMENTASI MODUL.....	43
5.1.	<i>Modul Bendahara</i>	43
5.2.	<i>Implementasi Antarmuka Pengguna</i>	120

BAB VI PENGUJIAN DAN EVALUASI	129
6.1. SKENARIO PENGUJIAN	129
6.1.1. <i>Melakukan pencatatan transaksi donasi</i>	<i>129</i>
6.1.2. <i>Melakukan pencaatatan penyaluran beasiswa.....</i>	<i>129</i>
6.1.3. <i>Mengupload dokumen berisi mutase rekening</i>	<i>129</i>
6.1.4. <i>Mengedit data transaksi yang tercatat.....</i>	<i>130</i>
6.1.5. <i>Membuat perencanaan penyaluran beasiswa</i>	<i>130</i>
6.1.6. <i>Menghapus data transaksi yang tercatat</i>	<i>130</i>
6.1.7. <i>Melihat perencanaan penyaluran beasiswa</i>	<i>130</i>
6.1.8. <i>Memecah transaksi menjadi beberapa</i>	<i>130</i>
6.1.9. <i>Melihat laporan kegiatan</i>	<i>131</i>
6.1.10. <i>Melihat laporan keuangan</i>	<i>131</i>
6.1.11. <i>Melihat neraca keuangan.....</i>	<i>131</i>
6.1.12. <i>Melakukan transfer antar akun.....</i>	<i>131</i>
6.1.13. <i>Membuat surat perintah penyaluran beasiswa</i>	<i>131</i>
6.2. EVALUASI PENGUJIAN	132
BAB VII KESIMPULAN DAN SARAN	133
DAFTAR PUSTAKA	135

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

GAMBAR 1.1 : JADWAL PELAKSANAAN PEMBANGUNAN SIBEATI ..	5
GAMBAR 4.1 : DIAGRAM USE CASE SIBEATI	17
GAMBAR 4.2 DIAGRAM AKTIVITAS UC001	32
GAMBAR 4.2 DIAGRAM AKTIVITAS UC002	33
GAMBAR 4.2 DIAGRAM AKTIVITAS UC003	34
GAMBAR 4.2 DIAGRAM AKTIVITAS UC004	35
GAMBAR 4.2 DIAGRAM AKTIVITAS UC005	36
GAMBAR 4.2 DIAGRAM AKTIVITAS UC006	36
GAMBAR 4.2 DIAGRAM AKTIVITAS UC007	37
GAMBAR 4.2 DIAGRAM AKTIVITAS UC008	38
GAMBAR 4.2 DIAGRAM AKTIVITAS UC009	39
GAMBAR 4.2 DIAGRAM AKTIVITAS UC010	39
GAMBAR 4.2 DIAGRAM AKTIVITAS UC011	40
GAMBAR 4.2 DIAGRAM AKTIVITAS UC012	41
GAMBAR 4.2 DIAGRAM AKTIVITAS UC013	42
GAMBAR 5.1 ANTARMUKA UC001	120
GAMBAR 5.2 ANTARMUKA UC002	121
GAMBAR 5.3 ANTARMUKA UC003	121
GAMBAR 5.4 ANTARMUKA UC004	122

GAMBAR 5.5 ANTARMUKA UC005	122
GAMBAR 5.6 ANTARMUKA UC006	123
GAMBAR 5.7 ANTARMUKA UC007	124
GAMBAR 5.8 ANTARMUKA UC008	124
GAMBAR 5.9 ANTARMUKA UC009	125
GAMBAR 5.10 ANTARMUKA UC010	126
GAMBAR 5.11 ANTARMUKA UC011	126
GAMBAR 5.12 ANTARMUKA UC012	127
GAMBAR 5.13 ANTARMUKA UC013	127

DAFTAR TABEL

TABEL 4.1 : KEBUTUHAN FUNGSIONAL.....	15
TABEL 4.2 :TABEL USE CASE UC001	18
TABEL 4.3 : TABEL USE CASE UC002.....	19
TABEL 4.4 : TABEL USE CASE UC003.....	20
TABEL 4.5 : TABEL USE CASE UC004.....	21
TABEL 4.6 : TABEL USE CASE UC005.....	22
TABEL 4.7 : TABEL USE CASE UC006.....	23
TABEL 4.8 : TABEL USE CASE UC007.....	24
TABEL 4.9 : TABEL USE CASE UC008.....	25
TABEL 4.10 : TABEL USE CASE UC009.....	26
TABEL 4.11 : TABEL USE CASE UC010.....	27
TABEL 4.12 : TABEL USE CASE UC011.....	28
TABEL 4.13 : TABEL USE CASE UC012.....	29
TABEL 4.14 : TABEL USE CASE UC013.....	30
TABEL 6.1 : TABEL EVALUASI PENGUJIAN APLIKASI SESUAI KEBUTUHAN	132

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Ikatan Alumni Teknik Informatika ITS memiliki satu unit kegiatan dalam pengelolaan beasiswa yang pedanaannya diperoleh dari donasi para alumni dan disalurkan kepada para mahasiswa yang masih aktif belajar di Teknik Informatika ITS. Ada beberapa ruang dalam tata kelola beasiswa ini yang bisa ditingkatkan melalui pemanfaatan teknologi informasi, sehingga beberapa kendala berikut dapat diatasi yaitu:

- a. Tata kelola keuangan masih manual
 - Dikelola dengan menggunakan excel
 - Tidak mudah untuk berbagi data karena harus berbagi file excel. Dan tidak semua data dalam file relevan dengan permintaan informasi yang diminta
 - Laporan keuangan masih harus dibagikan secara manual kepada para pemangku kepentingan melalui media sosial
- b. Tata kelola seleksi penerimaan beasiswa masih menggunakan Google Form
 - Calon penerima yang sudah pernah mendaftar harus memasukkan kembali semua data, meskipun ada data-data yang tidak berubah dari pendaftaran sebelumnya

- Data tidak terpusat. Salah satu anggota tim seleksi harus mengambil data dari google form lalu berbagi dengan anggota yang lain.
- Data tidak bisa langsung diperiksa oleh tim seleksi sesuai dengan peran masing-masing anggota.

Oleh karena itu, adanya sistem informasi yaitu SIBeATI ini adalah bentuk solusi untuk menciptakan sistem informasi terintegrasi yang akan meningkatkan kualitas tata kelola beasiswa alumni Teknik Informatika ITS.

1.2. Tujuan

Tujuan Kerja praktik kali ini adalah mengimplementasi-kan fitur-fitur berdasarkan kebutuhan SIBeATI yaitu:

- Membuat fitur yang memungkinkan mahasiswa untuk melakukan pendaftaran beasiswa
- Membuat fitur yang memungkinkan voter untuk memberikan suara terhadap data pendaftar beasiswa
- Membuat fitur yang memungkinkan tim seleksi untuk melakukan seleksi akhir terhadap data calon penerima beasiswa serta pengelolaan data berita dan beasiswa

1.3. Manfaat

Berikut manfaat yang diperoleh melalui kerja praktik dalam pembuatan fitur-fitur SIBeATI:

- Dapat menciptakan sistem informasi yang terintegrasi
- Dapat meningkatkan kualitas tata kelola beasiswa alumni Teknik Informatika ITS

1.4. Rumusan Permasalahan

Berikut rumusan masalah dalam pelaksanaan kerja praktik dalam pembuatan fitur-fitur SIBeATI:

- Bagaimana membangun fitur-fitur pada SIBeATI sebagai media tata kelola beasiswa alumni Teknik Informatika ITS yang efektif dan efisien?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik kali ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : Teknik Informatika ITS

Alamat: Jl. Teknik Kimia-Gedung Teknik Informatika Kampus ITS Surabaya Jalan Raya ITS, Sukolilo, Surabaya 60111

Waktu : 06 Juli 2020 – 28 Agustus 2020

1.6. Metodologi Kerja Praktik

1. Perumusan Masalah

Untuk mengetahui domain dan fungsionalitas, dijelaskan secara rinci bagaimana sistem yang harus dibuat. Penjelasan oleh pembimbing lapangan kerja praktik kali ini menghasilkan beberapa catatan mengenai gambaran secara garis besar tentang kebutuhan atau fitur apa saja yang harus ada di dalam

sistem informasi. Setelah mendapatkan gambaran sistem, diskusi lebih lanjut dilakukan guna menentukan rancangan serta *tools* pendukung pembuatan sistem informasi.

2. Studi Literatur

Pada tahap ini, setelah ditentukannya rancangan *database*, bahasa pemrograman sampai dengan teknologi beserta *tools* tambahan yang digunakan, dilakukan studi literatur lanjut mengenai bagaimana penggunaannya dalam membangun sistem informasi sesuai dengan yang diharapkan.

Untuk konstruksi SIBeATI menggunakan 2 environment yaitu production environment (its.ac.id/informatika/beasiswa) dan testing environment (its.ac.id/informatika/beasiswa-test), dimana developing dan testing dilakukan di testing environment sementara hasil akhir yang telah diuji dilakukan di production environment. Bahasa pemrograman yang digunakan adalah PHP dengan framework yaitu CodeIgniter (menggunakan arsitektur multi module MVC). Sementara untuk database yang digunakan adalah MySQL.

3. Analisis dan Perancangan Sistem

Langkah ini meliputi penjelasan awal tentang sistem informasi. Bagaimana cara kerja sistem informasi dengan skenario tertentu. Dari penjelasan awal telah didapatkan beberapa kebutuhan fungsional secara garis besar. Kemudian dilanjutkan dengan

memperjelas dan menspesifikkan kebutuhan-kebutuhan tersebut. Dilanjutkan berdiskusi dengan pembimbing lapangan untuk mengetahui apakah kebutuhan-kebutuhan tersebut sudah tepat.

4. Implementasi Sistem

Implementasi sistem informasi didasarkan oleh perancangan dan analisis sebelumnya. Penentuan atribut atau fitur yang akan digunakan pada model juga didasari pada analisis sebelumnya. Penentuan tipe data dan format keluaran juga disesuaikan dengan kebutuhan.

Pengerjaan dilakukan dengan penyampaian progress setiap minggu, dengan setiap minggunya menargetkan perkembangan dari hari sebelumnya. Penyampaian progress dilakukan setiap hari minggu pukul 08.00-09.00 WIB melalui aplikasi Zoom. Selain itu pembimbing lapangan juga memberikan masukan-masukan apabila terdapat kendala. Berikut adalah jadwal pembuatan sistem informasi tersebut:

Jadwal Proyek SIBeATI

Activity	19-Jun-2020	26-Jun-2020	3-Jul-2020	10-Jul-2020	17-Jul-2020	24-Jul-2020	31-Jul-2020	7-Aug-2020	14-Aug-2020	21-Aug-2020	28-Aug-2020
Initiation											
Requirement & Design											
Build											
Training & Test											
Implementation											
Closing											

Gambar 1.1 : Jadwal Pelaksanaan Pembangunan SIBeATI

5. Pengujian dan Evaluasi

Pengujian dilakukan oleh pembimbing lapangan dan anggota alumni Teknik Informatika ITS yang lain. Pengujian dilakukan terhadap setiap fitur yang sudah selesai untuk memberikan evaluasi ketika ada yang tidak sesuai, dan persetujuan apabila sudah sesuai.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari 7 bab dengan rincian sebagai berikut :

1. Bab I: Pendahuluan

Bab ini berisi tentang latar belakang masalah, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

2. Bab II: Profil Instansi

Bab ini berisi sekilas tentang profil PT. Syarihubs Indoensia.

3. Bab III: Tinjauan Pustaka

Dalam bab ini dibahas mengenai konsep-konsep pembuatan model, dasar teori, teknologi yang dipakai dalam pembuatan model.

4. Bab IV: Analisis dan Perancangan Sistem

Dalam bab ini dibahas tentang proses analisa kebutuhan berdasarkan kondisi yang sesungguhnya dan perancangannya yang meliputi desain aplikasi

yang akan dikembangkan. Proses analisa dan desain aplikasi menghasilkan daftar fitur yang dibutuhkan.

5. Bab V: Desain Model dan Implementasi Sistem

Dalam bab ini dibahas tentang desain model dan implementasi secara keseluruhan.

6. Bab VI: Pengujian dan Evaluasi

Dalam bab ini dibahas tentang skenario pengujian, dan evaluasi pengujian setelah model selesai dibangun.

7. Bab VII: Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan dan saran yang didapatkan dari tugas selama kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL INSTANSI

2.1. Profil Instansi

Ikatan alumni Teknik Informatika ITS adalah wadah para alumni departemen Teknik Informatika ITS (Institut Teknologi Sepuluh Nopember) Surabaya untuk berkelompok, berorganisasi, dan menyampaikan aspirasinya. Sebagai elemen yang tidak bisa dipisahkan dengan almamaternya yaitu Teknik Informatika ITS, Ikatan Alumni memiliki perencanaan strategis dalam upaya melayani anggotanya. Dimana anggotanya adalah seluruh lulusan S1 Teknik Informatika ITS Surabaya tanpa terkecuali. Dalam pelayanan tersebut Ikatan Alumni menitik beratkan agar para alumni mempunyai daya saing yang tinggi di tingkat nasional maupun internasional.

Ikatan Alumni Teknik Informatika ITS memiliki satu unit kegiatan dalam pengelolaan beasiswa yang pendanaannya diperoleh dari donasi para alumni dan disalurkan kepada para mahasiswa yang masih aktif belajar di Teknik Informatika ITS. Terdapat dua ruang dalam tata kelola beasiswa ini yaitu tata kelola keuangan (menangani pembukuan dan mengelola data-data transaksi donasi serta penyaluran beasiswa) dan tata kelola kepesertaan (menangani pendaftaran dan proses seleksi penerimaan beasiswa).

Namun, tata kelola beasiswa selama ini masih berjalan secara manual yang seharusnya bisa ditingkatkan melalui pemanfaatan teknologi informasi. Sehingga saat ini Ikatan Alumni Teknik Informatika beasiswa mengembangkan sistem informasi yang dapat meningkatkan kualitas tata kelola beasiswa alumni Teknik Informatika ITS yaitu SIBeATI (Sistem Informasi Beasiswa Alumni Teknik Informatika ITS).

BAB III

TINJAUAN PUSTAKA

3.1. Sistem Informasi

Sistem informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Tujuan dari sistem informasi adalah menghasilkan informasi. Data diolah menjadi bentuk yang berguna bagi para pemakainya serta harus didukung oleh tiga pilar yaitu tepat kepada orangnya atau relevan (*relevance*), tepat waktu (*timeliness*), dan tepat nilainya atau akurat (*accurate*).

3.2. PHP

PHP adalah sebuah bahasa pemrograman yang berjalan dalam sebuah *web-server* (*serverside*). PHP pada dasarnya merupakan singkatan dari PHP : Hypertext Preprocessor. Dalam praktiknya PHP biasanya digunakan bersama dengan penggunaan bahasa pemrograman lainnya seperti bahasa pemrograman HTML dan bahasa pemrograman JavaScript.

3.3. MySQL

MySQL adalah sebuah database manajemen sistem (DBMS) populer yang memiliki fungsi sebagai *Relational Database Manajemen System* (RDBMS). Selain itu MySQL *software* merupakan suatu aplikasi

yang sifatnya *open source* serta server basis data MySQL memiliki kinerja sangat cepat, *reliable*, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems*. Dikarenakan faktor open source dan populer tersebut maka cocok untuk mendemonstrasikan proses replikasi basis data.

3.5. HMVC

HMVC merupakan hirarki atau susunan dari pola MVC yang terusun menjadi satu kesatuan aplikasi. HMVC pada dasarnya merupakan singkatan dari Hierarchical-Model-View-Controller. Setiap susunan MVC benar-benar independen (berdiri sendiri) dan dapat mengeksekusi tanpa kehadiran yang lain. Semua permintaan dari susunan masing-masing MVC harus menggunakan antarmuka (*view*) dan pengendali (*controller*) sendiri dan tidak pernah memuat model atau perpustakaan di luar domain atau susunan MVC sendiri.

3.6. CodeIgniter

CodeIgniter adalah *framework* aplikasi berbasis web yang mengusung model MVC (Model, View, Controller) atau HMVC (Hierarchical-Model-View-Controller). Salah satu *framework* PHP ini dapat menjadi *tools* bagi seorang *web developer* untuk mengembangkan suatu situs dengan lebih mudah. CodeIgniter bersifat *open source* (tidak berbayar)

sehingga *framework* PHP ini dapat dengan mudah didapatkan.

3.8. Visual Studio Code

Visual Studio Code adalah aplikasi editor kode yang di kembangkan oleh microsoft . Visual Studio Code dapat berjalan di platform Windows, Linux dan MacOS. Visual Studio Code merupakan salah satu kode editor yang banyak di gunakan dikarenakan tampilannya yang sederhana serta memiliki fitur yang cukup lengkap.

[Halaman ini sengaja dikosongkan]

BAB IV

ANALISIS DAN DESAIN

4.1. Analisis Sistem

4.1.1. Definisi Umum Fitur

Dalam SIBeATI terdapat 5 stakeholder yaitu user umum, mahasiswa, bendahara, voter, dan tim seleksi dimana user umum dan bendahara masuk ke tata kelola keuangan sementara mahasiswa, voter, dan tim seleksi masuk ke tata kelola kepesertaan. Kami kebagian tugas untuk membuat system pada bagian tata kelola keuangan.

4.1.2. Analisis Kebutuhan Fungsional

Kebutuhan fungsional yang diperlukan pada SIBeATI dapat dilihat pada tabel 4.1.

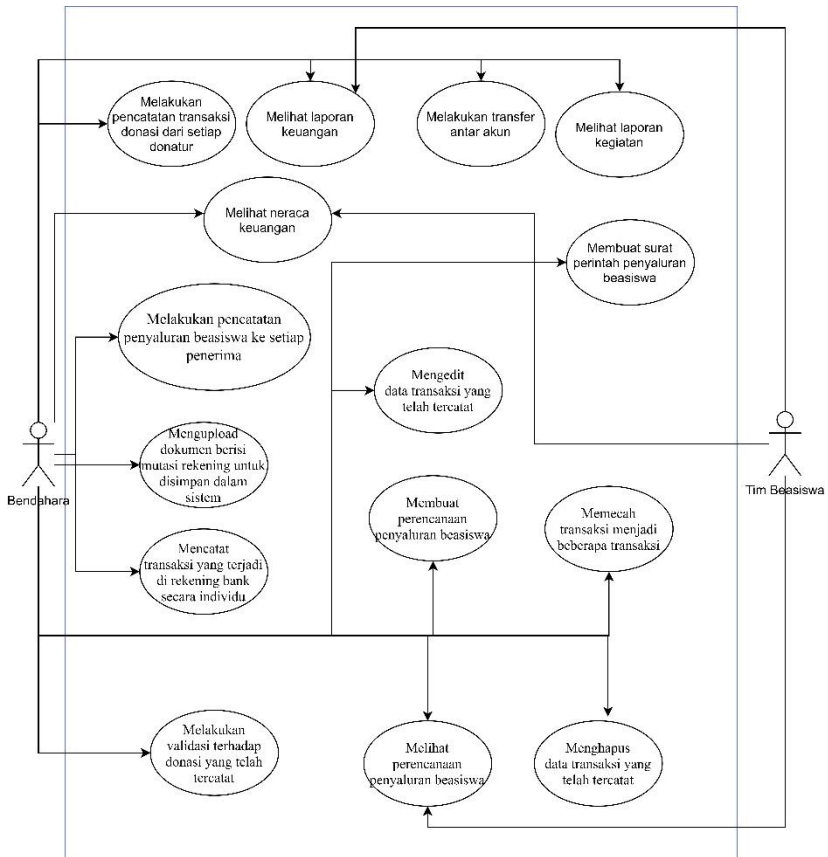
Tabel 4.1 : Kebutuhan Fungsional

Kode Kebutuhan	Deskripsi Kebutuhan
FS-001	Melakukan pencatatan transaksi donasi dari setiap donatur
FS-002	Melakukan pencatatan penyaluran beasiswa ke setiap penerima
FS-003	Mengupload dokumen berisi mutasi rekening untuk disimpan dalam sistem
FS-004	Mengedit data transaksi yang telah tercatat
FS-005	Membuat perencanaan penyaluran beasiswa

FS-006	Menghapus data transaksi yang telah tercatat
FS-007	Melihat perencanaan penyaluran beasiswa
FS-008	Memecah transaksi menjadi beberapa transaksi
FS-009	Melihat laporan kegiatan
FS-010	Melihat laporan keuangan
FS-011	Melihat neraca keuangan
FS-012	Melakukan transfer antar akun
FS-013	Membuat surat perintah penyaluran beasiswa

4.2. Diagram Kasus Penggunaan

Pembahasan dengan pembimbing lapangan tentang fitur-fitur yang diperlukan dalam SIBeATI menghasilkan beberapa fitur yang dijadikan diagram kasus penggunaan (Use Case Diagram) sehingga memudahkan untuk dipahami. Use Case Diagram yang telah dibuat dapat dilihat pada Gambar 4.1.



Gambar 4.1 : Diagram Use Case SIBeATI

4.3. Spesifikasi Kasus Penggunaan

4.3.1. Melakukan Pencatatan Transaksi Donasi

Tabel 4.2 :Tabel Use Case UC001

Nama	Melakukan Pendaftaran Akun
Kode	UC001
Deskripsi	Bendahara dapat mencatat transaksi donasi dari tiap donatur
Tipe	Fungsional
Pemicu	Bendahara menekan tombol save pada edit data transaksi
Aktor	Bendahara
Kondisi Awal	Transaksi baru saja di upload belum di edit sama sekali
Kondisi Akhir	Transaksi tercatat sebagai donasi dari seorang donatur
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman buku besar 2. Bendahara menekan tombol edit pada transaksi donasi 3. Bendahara mengedit data transaksi sesuai berikut <ol style="list-style-type: none"> a. Jenis transaksi: Penerimaan b. Akun: Beasiswa c. Kategori: Donasi d. Inisial donatur: <donatur> 4. Bendahara menekan tombol save

Alur Kejadian Alternatif	4a. Data yang dimasukkan tidak valid - Sistem tidak menyimpan perubahan data
Pengecualian	-

4.3.2. Melakukan Pencatatan Penyaluran Beasiswa

Tabel 4.3 : Tabel Use Case UC002

Nama	Melakukan Login
Kode	UC002
Deskripsi	Bendahara dapat melakukan pencatatan penyaluran beasiswa
Tipe	Fungsional
Pemicu	Bendahara menekan tombol save pada edit transaksi
Aktor	Bendahara
Kondisi Awal	Transaksi belum tercatat sebagai penyaluran beasiswa
Kondisi Akhir	Transaksi tercatat sebagai penyaluran beasiswa
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman buku besar 2. Bendahara menekan tombol edit pada transaksi donasi 3. Bendahara mengedit data transaksi sesuai berikut

	a. Jenis transaksi: Pengeluaran b. Akun: Beasiswa c. Kategori: Penyaluran 4. Bendahara menekan tombol save
Alur Kejadian Alternatif	4a. Data yang dimasukkan tidak valid - Sistem tidak menyimpan perubahan data
Pengecualian	-

4.3.3. Mengupload Dokumen Mutasi Rekening

Tabel 4.4 : Tabel Use Case UC003

Nama	Mengupload Dokumen Mutasi Rekening
Kode	UC003
Deskripsi	Bendahara dapat mengupload mutasi rekening untuk disimpan pada database
Tipe	Fungsional
Pemicu	Bendahara menekan tombol upload
Aktor	Bendahara
Kondisi Awal	Data mutasi rekening belum tersimpan
Kondisi Akhir	Data mutase rekening tersimpan

Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman upload mutasi 2. Bendahara menekan tombol upload 3. Bendahara memilih data yang ingin diupload 4. Bendahara menekan tombol upload
Alur Kejadian Alternatif	4a. File yang di upload tidak sesuai - Sistem memberi tahu error nya
Pengecualian	-

4.3.4. Mengedit Data Transaksi Yang Tercatat

Tabel 4.5 : Tabel Use Case UC004

Nama	Mengedit data transaksi yang tercatat
Kode	UC004
Deskripsi	Bendahara dapat mengedit data transaksi yang tercatat
Tipe	Fungsional
Pemicu	Bendahara menekan tombol save
Aktor	Bendahara
Kondisi Awal	Data transaksi belum dirubah
Kondisi Akhir	Data transaksi telah berubah

Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman buku besar 2. Bendahara menekan tombol edit di transaksi yang ingin di rubah 3. Bendahara mengisi data perubahan pada form yang disediakan 4. Bendahara menekan tombol save
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 4a. Data yang diisi tidak valid <ul style="list-style-type: none"> - Sistem tidak menyimpan data perubahan
Pengecualian	-

4.3.5. Membuat Perencanaan Penyaluran Beasiswa

Tabel 4.6 : Tabel Use Case UC005

Nama	Membuat perencanaan penyaluran beasiswa
Kode	UC005
Deskripsi	Bendahara dapat membuat perencanaan penyaluran beasiswa
Tipe	Fungsional
Pemicu	Bendahara menekan tombol simpan pada halaman perencanaan beasiswa
Aktor	Bendahara

Kondisi Awal	Bendahara belum membuat perencanaan penyaluran beasiswa
Kondisi Akhir	Perencanaan penyaluran beasiswa tersimpan
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman perencanaan beasiswa 2. Bendahara menekan tombol buat perencanaan baru 3. Bendahara memasukkan data sesuai form yang tersedia 4. Sistem menampilkan hasil kalkulasi perencanaan 5. Bendahara memasukkan data perencanaan untuk periode berapa 6. Bendahara menekan tombol simpan
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.6. Menghapus Data Transaksi Yang Tercatat

Tabel 4.7 : Tabel Use Case UC006

Nama	Menghapus data transaksi yang tercatat
Kode	UC006

Deskripsi	Bendahara dapat menghapus data transaksi yang sudah tercatat
Tipe	Fungsional
Pemicu	Bendahara menekan tombol delete
Aktor	Bendahara
Kondisi Awal	Data transaksi belum terhapus
Kondisi Akhir	Data transaksi sudah terhapus
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman buku besar 2. Bendahara menekan tombol delete pada transaksi yang ingin dihapus
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.7. Melihat Perencanaan Penyaluran Beasiswa

Tabel 4.8 : Tabel Use Case UC007

Nama	Melihat perencanaan penyaluran beasiswa
Kode	UC007

Deskripsi	Bendahara dan tim beasiswa dapat melihat perencanaan penyaluran beasiswa
Tipe	Fungsional
Pemicu	Bendahara pergi ke halaman perencanaan beasiswa
Aktor	Bendahara, Tim Beasiswa
Kondisi Awal	Belum berada di halaman perencanaan beasiswa
Kondisi Akhir	Berada di halaman perencanaan beasiswa
Alur Kejadian Secara Normal	1. Bendahara menekan tombol perencanaan pada sidebar
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.8. Memecah Transaksi Menjadi Banyak

Tabel 4.9 : Tabel Use Case UC008

Nama	Memecah transaksi menjadi banyak
Kode	UC008

Deskripsi	Bendahara dapat memecah transaksi menjadi beberapa transaksi
Tipe	Fungsional
Pemicu	Bendahara menekan tombol pecah
Aktor	Bendahara
Kondisi Awal	Data transaksi belum terpecah
Kondisi Akhir	Data transaksi terpecah jadi beberapa transaksi
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman buku besar 2. Bendahara menekan tombol pecah di transaksi yang ingin di pecah 3. Bendahara mengisi rincian pemecahan sesuai form yang ada 4. Bendahara menekan tombol pecah
Alur Kejadian Alternatif	4a. Form pemecahan tidak valid - Sistem tidak jadi memecah
Pengecualian	-

4.3.9. Melihat Laporan Kegiatan

Tabel 4.10 : Tabel Use Case UC009

Nama	Melihat Laporan Kegiatan
Kode	UC009
Deskripsi	Bendahara dapat melihat laporan kegiatan
Tipe	Fungsional
Pemicu	Bendahara menekan tombol laporan
Aktor	Bendahara
Kondisi Awal	Laporan belum ditampilkan
Kondisi Akhir	Laporan sudah ditampilkan
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman laporan 2. Bendahara memilih tipe laporan ringkasan 3. Bendahara memilih periode yang ingin ditampilkan 4. Sistem menampilkan laporan pada periode tersebut
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.10. Melihat Laporan Keuangan

Tabel 4.11 : Tabel Use Case UC010

Nama	Melihat Laporan Keuangan
Kode	UC010
Deskripsi	Bendahara dapat melihat laporan keuangan
Tipe	Fungsional
Pemicu	Bendahara menekan tombol rekapitulasi
Aktor	Bendahara
Kondisi Awal	Laporan belum ditampilkan
Kondisi Akhir	Laporan sudah ditampilkan
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara menekan tombol rekapitulasi pada sidebar 2. Sistem menampilkan laporan rekapitulasi keuangan
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.11. Melihat Neraca Keuangan

Tabel 4.12 : Tabel Use Case UC011

Nama	Melihat Laporan Kegiatan
Kode	UC011
Deskripsi	Bendahara dapat melihat neraca keuangan

Tipe	Fungsional
Pemicu	Bendahara menekan tombol laporan
Aktor	Bendahara
Kondisi Awal	Neraca belum ditampilkan
Kondisi Akhir	Neraca sudah ditampilkan
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman laporan 2. Bendahara memilih tipe laporan neraca 3. Sistem menampilkan neraca
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.12. Melakukan Transfer Antar Akun

Tabel 4.13 : Tabel Use Case UC012

Nama	Melakukan transfer antar akun
Kode	UC012
Deskripsi	Bendahara dapat mentransfer dari akun satu ke akun lainnya
Tipe	Fungsional
Pemicu	Bendahara menekan transfer
Aktor	Bendahara

Kondisi Awal	Transaksi masih berada di akun lama
Kondisi Akhir	Transaksi berada di akun lain, ada transaksi baru untuk menyeimbangkan neraca
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman buku besar 2. Bendahara menekan tombol transfer pada transaksi yang ingin di transfer 3. Bendahara memilih ingin mentransfer ke akun mana dan juga mengisi deskripsi sesuai form 4. Bendahara menekan tombol transfer 5. Sistem mentransfer transaksi ke akun tujuan dan membuat transaksi baru untuk menyeimbangkan neraca
Alur Kejadian Alternatif	-
Pengecualian	-

4.3.13. Membuat Surat Perintah Penyaluran Beasiswa

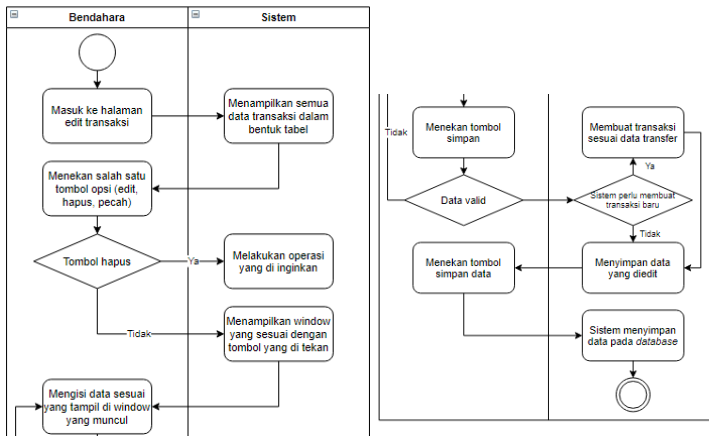
Tabel 4.14 : Tabel Use Case UC013

Nama	Membuat surat perintah penyaluran beasiswa
Kode	UC013
Deskripsi	Bendahara dapat membuat surat perintah penyaluran beasiswa pada suatu periode
Tipe	Fungsional
Pemicu	Bendahara menekan tombol print surat perintah
Aktor	Bendahara
Kondisi Awal	Surat perintah penyaluran belum dibuat
Kondisi Akhir	Surat perintah penyaluran terbuat
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Bendahara masuk ke halaman surat perintah 2. Bendahara memilih ingin membuat surat perintah untuk periode berapa 3. Sistem menampilkan surat perintah siap untuk di print langsung dan bisa di simpan
Alur Kejadian Alternatif	-
Pengecualian	-

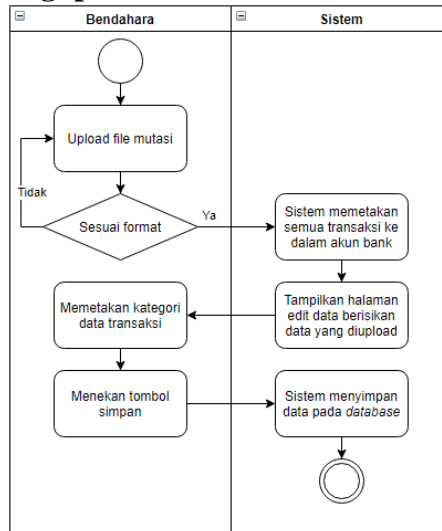
4.4. Diagram Aktivitas

4.4.1. Melakukan pencatatan transaksi donasi

Gambar 4.2 Diagram Aktivitas UC001

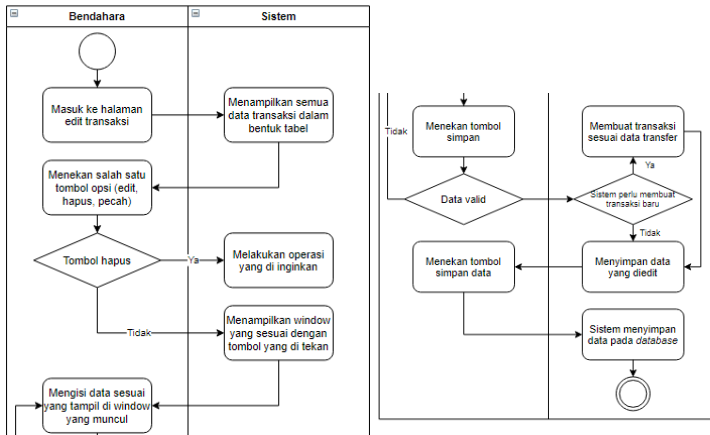


4.4.3. Mengupload dokumen berisi mutasi rekening



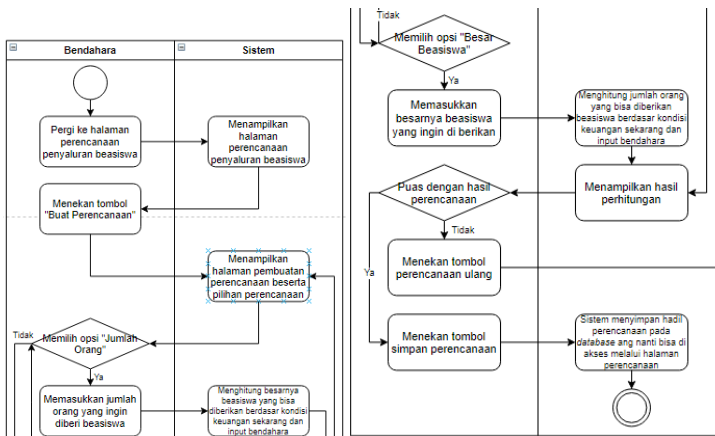
Gambar 4.2 Diagram Aktivitas UC003

4.4.4. Mengedit data transaksi



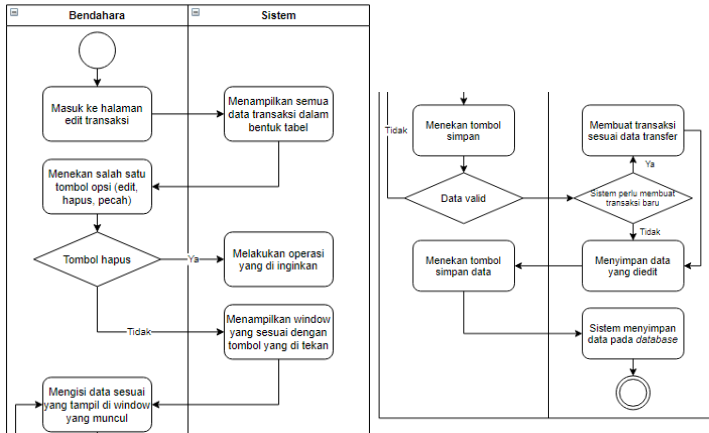
Gambar 4.2 Diagram Aktivitas UC004

4.4.5. Membuat perencanaan penyaluran beasiswa



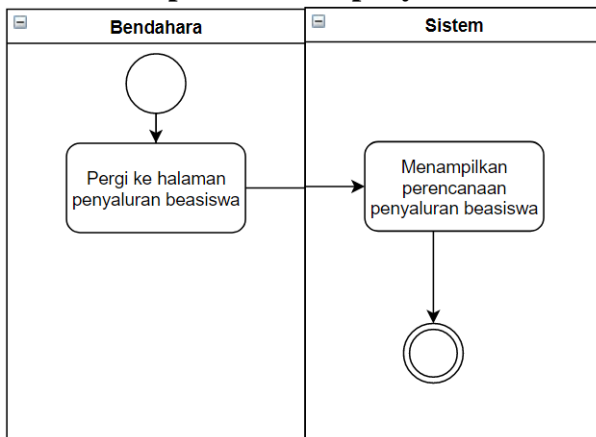
Gambar 4.2 Diagram Aktivitas UC005

4.4.6. Menghapus data transaksi



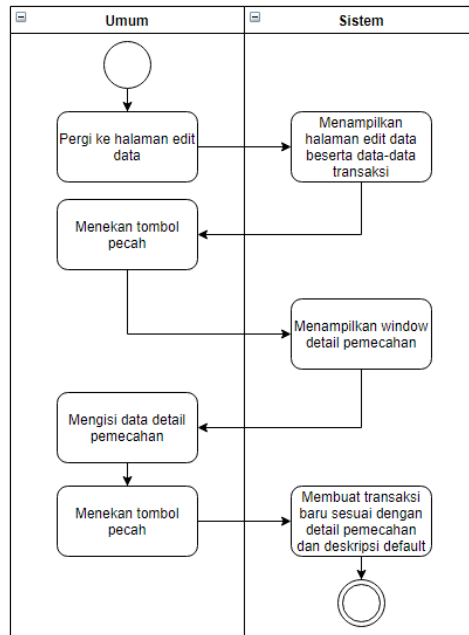
Gambar 4.2 Diagram Aktivitas UC006

4.4.7. Melihat perencanaan penyaluran



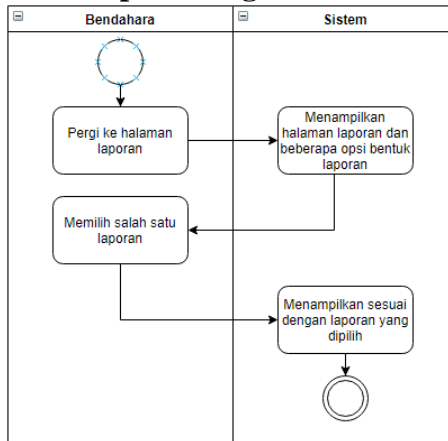
Gambar 4.2 Diagram Aktivitas UC007

4.4.8. Memecah transaksi menjadi beberapa



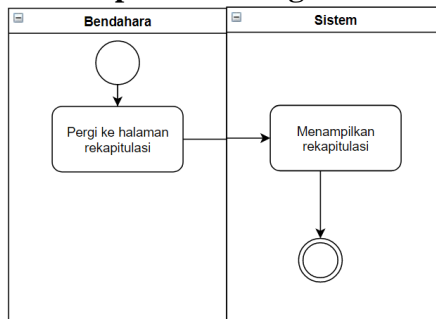
Gambar 4.2 Diagram Aktivitas UC008

4.4.9. Melihat laporan kegiatan



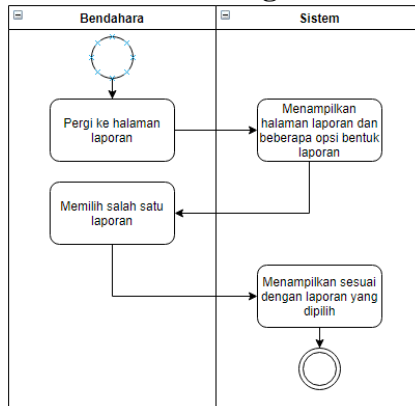
Gambar 4.2 Diagram Aktivitas UC009

4.4.10. Melihat laporan keuangan



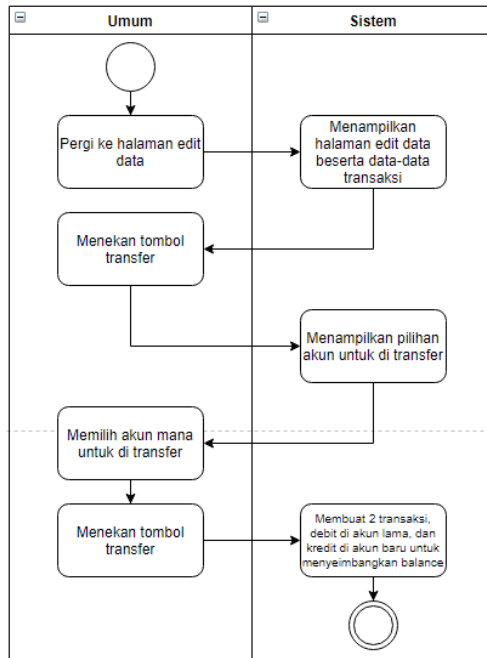
Gambar 4.2 Diagram Aktivitas UC010

4.4.11. Melihat neraca keuangan



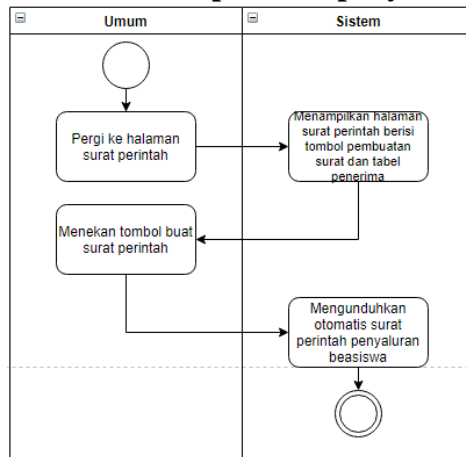
Gambar 4.2 Diagram Aktivitas UC011

4.4.12. Melakukan transfer antar akun



Gambar 4.2 Diagram Aktivitas UC012

4.4.13. Membuat surat perintah penyaluran beasiswa



Gambar 4.2 Diagram Aktivitas UC013

BAB V

IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari sistem informasi yang telah dibuat. Implementasi ini akan dibagi ke dalam beberapa bagian, yaitu bagian implementasi modul dan implementasi antarmuka pengguna.

5. Implementasi Modul

Implementasi modul ini berisi logika yang digunakan sistem informasi yang terdiri dari model dan controller.

5.1. Modul Bendahara

5.1.1. Controllers

5.1.1.1. CRUD akun

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Crud_ak extends CI_Controller {
    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        $this->session->set_userdata('prev_url', current_url());
    }
}
```

```

$this->load->model('akun');
$this->load->model('kategori');

$allAkun = $this->akun->getAll();
$allKategori = $this->kategori->getAll();

$data = array(
    'allAkun' => $allAkun,
    'allKategori' => $allKategori
);

$this->template-
>load('dashboard/template', 'crudAK', 'CRUD AK', $data);
}

public function createKategori(){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('kategori');
        $data = $this->input->post(null, TRUE);

        if(isset($data['nama-kt'])){
            $this->kategori->create($data['nama-kt']);
        }
    }
}

```

```

        redirect($this->session->userdata('prev_url'));
    }
}

public function createAkun(){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('akun');
        $data = $this->input->post(null, TRUE);

        if(isset($data['nama-ak'])){
            $this->akun->create($data['nama-ak']);

            redirect($this->session->userdata('prev_url'));
        }
    }
}

public function deleteAkun($id){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>

```

```

        alert('Maaf anda bukan bendahara');
        window.location="'.site_url('dashboard')."';
    </script>";
}
else{
    $this->load->model('akun');

    $this->akun->delete($id);

    redirect($this->session->userdata('prev_url'));
}
}

public function deleteKategori($id){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('kategori');

        $this->kategori->delete($id);

        redirect($this->session->userdata('prev_url'));
    }
}
}

```

```
}

```

5.1.1.2. CRUD donatur

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Crud_donatur extends CI_Controller {
    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        $this->session->set_userdata('prev_url', current_url());

        $this->load->model('donatur');

        $allDonatur = $this->donatur->getAll();

        $data = array(
            'allDonatur' => $allDonatur
        );

        $this->template-
    >load('dashboard/template', 'crudDonatur', 'CRUD Donatur', $data);

```



```

    }

    public function save($inisial){
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }
        else{
            $this->load->model('donatur');
            $data = $this->input->post(null, TRUE);

            if(isset($data['nama'])){
                $param = array(
                    'inisial' => $inisial,
                    'nama' => $data['nama']
                );

                $this->donatur->save($param);

                redirect($this->session->userdata('prev_url'));
            }
        }
    }

    public function delete($inisial){
        if($this->session->userdata('role') != "bendahara"){

```

```

        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('donatur');

        $this->donatur->delete($inisial);

        redirect($this->session->userdata('prev_url'));
    }
}
}
}

```

5.1.1.3. CRUD periode

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Crud_donatur extends CI_Controller {

    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }
    }
}

```

```

$this->session->set_userdata('prev_url', current_url());

$this->load->model('donatur');

$allDonatur = $this->donatur->getAll();

$data = array(
    'allDonatur' => $allDonatur
);

$this->template-
>load('dashboard/template', 'crudDonatur', 'CRUD Donatur', $data);
}

public function save($inisial){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('donatur');
        $data = $this->input->post(null, TRUE);

        if(isset($data['nama'])){
            $param = array(

```

```

        'inisial' => $inisial,
        'nama' => $data['nama']
    );

    $this->donatur->save($param);

    redirect($this->session->userdata('prev_url'));
}
}

public function delete($inisial){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('donatur');

        $this->donatur->delete($inisial);

        redirect($this->session->userdata('prev_url'));
    }
}
}

```

5.1.1.4. Edit transaksi

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Edit extends CI_Controller {

    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        $this->session->set_userdata('prev_url', current_url());

        $this->load->model('transaksi');

        $keys = array('Tanggal', 'Deskripsi', 'Debit', 'Kredit', 'Saldo',
            'Periode', 'Jenis Transaksi', 'Akun', 'Kategori', 'Alias Donatur');

        $postData = $this->input->post(null, TRUE);

        if(isset($postData['search'])){
            $searchData = array(
                'deskripsi_s' => '%' . $postData['deskripsi_s'] . '%',
                'debit_s' => $postData['debit_s'] == '' ? '%' : $postData
            ['debit_s'],

```

```

        'kredit_s' => $postData['kredit_s'] == '' ? '%' : $postData['kredit_s'],
        'saldo_s' => $postData['saldo_s'] == '' ? '%' : $postData['saldo_s'],
        'periode_s' => '%' . $postData['periode_s'] . '%',
        'tanggal_s' => '%' . $postData['tanggal_s'] . '%',
        'id_jenis_s' => '%' . $postData['id_jenis_s'] . '%',
        'id_kategori_s' => '%' . $postData['id_kategori_s'] . '%',
        'id_akun_s' => '%' . $postData['id_akun_s'] . '%',
        'donatur_s' => '%' . $postData['donatur_s'] . '%'
    );

    $datas = $this->transaksi->search($searchData);
}
else{
    $datas = $this->transaksi->getAll();
}

$akuns = $this->transaksi->getAkun();
$kategoris = $this->transaksi->getKategori();
$jenis_trans = $this->transaksi->getJenisTrans();

$data = array(
    'keys' => $keys,
    'datas' => $datas,
    'akuns' => $akuns,
    'kategoris' => $kategoris,
    'jenis_trans' => $jenis_trans

```

```

    );

    $this->template-
>load("dashboard/template", "edit/edit", "Edit Data", $data);
    }

    public function editAwal(){
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        $this->session->set_userdata('prev_url', current_url());

        $this->load->model('temp_transaksi');

        $keys = array('Tanggal', 'Deskripsi', 'Debit', 'Kredit', 'Saldo',
'Periode', 'Jenis Transaksi', 'Akun', 'Kategori', 'Alias Donatur');
        $datas = $this->temp_transaksi->getAll();

        $akuns = $this->temp_transaksi->getAkun();
        $kategoris = $this->temp_transaksi->getKategori();
        $jenis_trans = $this->temp_transaksi->getJenisTrans();

        $data = array(
            'keys' => $keys,

```

```

        'datas' => $datas,
        'akuns' => $akuns,
        'kategoris' => $kategoris,
        'jenis_trans' => $jenis_trans
    );

    $this->template-
>load("dashboard/template", "edit/editAwal", "Edit Data", $data);
    }

    public function save(){
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        if($this->session-
>userdata('prev_url') == site_url('bendahara/edit/editAwal')){
            $this->load->model('temp_transaksi', 'transaksi');
        }
        else if($this->session-
>userdata('prev_url') == site_url('bendahara/edit')){
            $this->load->model('transaksi');
        }
        else{
            echo "<script>

```



```

        alert('Mohon akses dari halaman yang benar');
        window.location="'.site_url('dashboard')."';
    </script>";
}

$data = $this->input->post(null, TRUE);

if(isset($data['id'])){
    $result = $this->transaksi->save($data);
    if($result != 0){
        redirect($this->session->userdata('prev_url'));
    }
    else{
        //gagal
        echo "<script>
            alert('Maaf, edit gagal');
            window.location='" . $this->session->
>userdata('prev_url') . "';
            </script>";
    }
}

}

public function delete($data){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';

```

```

        </script>";
    }

    if($this->session->userdata('prev_url') == site_url('bendahara/edit/editAwal')){
        $this->load->model('temp_transaksi', 'transaksi');
    }
    else if($this->session->userdata('prev_url') == site_url('bendahara/edit')){
        $this->load->model('transaksi');
    }
    else{
        echo "<script>
            alert('Mohon akses dari halaman yang benar');
            window.location='".site_url('dashboard')."';
        </script>";
    }

    if($data != NULL){
        $result = $this->transaksi->delete($data);
        if($result != 0){
            redirect($this->session->userdata('prev_url'));
        }
        else{
            //gagal
            echo "<script>
                alert('Maaf, hapus gagal');
            </script>";
        }
    }
}

```

```

        window.location='".$this->session-
>userdata('prev_url')."';
        </script>";
    }
}

public function pecah(){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".$site_url('dashboard')."';
        </script>";
    }

    if($this->session-
>userdata('prev_url') == site_url('bendahara/edit/editAwal')){
        $this->load->model('temp_transaksi', 'transaksi');
    }

    else if($this->session-
>userdata('prev_url') == site_url('bendahara/edit')){
        $this->load->model('transaksi');
    }

    else{
        echo "<script>
            alert('Mohon akses dari halaman yang benar');
            window.location='".$site_url('dashboard')."';
        </script>";
    }
}

```

```

    }

    $data = $this->input->post(null, TRUE);

    if(isset($data['pc-id'])){
        // cek debit kredit apakah valid
        $totalDebit = '0';
        $totalKredit = '0';

        foreach($data['pc-debit'] as $debit){
            $totalDebit = $totalDebit + $debit;
        }

        foreach($data['pc-kredit'] as $kredit){
            $totalKredit = $totalKredit + $kredit;
        }

        if($totalDebit != $data['pc-debit_awal']){
            echo "<script>
                alert('Maaf, data total debit tidak sesuai data awal');
                window.location='" . $this->session->userdata('prev_url') . "';
            </script>";
        }

        else if($totalKredit != $data['pc-kredit_awal']){
            echo "<script>
                alert('Maaf, data total kredit tidak sesuai data awal');
                window.location='" . $this->session->userdata('prev_url') . "';
            </script>";
        }
    }
}

```

```

        </script>";
    }

    $count = 0;
    // simpan
    foreach($data['pc-deskripsi'] as $nothing){
        $param = array(
            'deskripsi' => $data['pc-deskripsi'][$count],
            'debit' => $data['pc-debit'][$count],
            'kredit' => $data['pc-kredit'][$count],
            'kategori' => $data['pc-kategori'],
            'akun' => $data['pc-akun'],
            'periode' => $data['pc-periode'],
            'tanggal' => $data['pc-tanggal'],
            'inisial_donatur' => $data['pc-
inisial_donatur'][$count]
        );

        $this->transaksi->split($param);
        $count++;
    }

    $this->transaksi->delete($data['pc-id']);
    redirect($this->session->userdata('prev_url'));
}

}

public function transfer(){

```

```

        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        if($this->session->userdata('prev_url') == site_url('bendahara/edit/editAwal')){
            $this->load->model('temp_transaksi', 'transaksi');
        }
        else if($this->session->userdata('prev_url') == site_url('bendahara/edit')){
            $this->load->model('transaksi');
        }
        else{
            echo "<script>
                alert('Mohon akses dari halaman yang benar');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        $data = $this->input->post(null, TRUE);
        if(isset($data['tf-id'])){
            $param = array(
                'deskripsi' => $data['tf-deskripsi'],
                'debit' => $data['tf-debit'],
                'kredit' => $data['tf-kredit'],
            );
        }
    }
}

```

```

        'kategori' => $data['tf-kategori'],
        'akun' => $data['tf-akun'],
        'akun_tujuan' => $data['tf-akun_tujuan'],
        'periode' => $data['tf-periode'],
        'tanggal' => $data['tf-tanggal'],
        'inisial_donatur' => $data['tf-inisial_donatur']
    );

    $this->transaksi->createBalance($param);
    $this->transaksi->transfer($param);

    redirect($this->session->userdata('prev_url'));
}

}

public function saveDatabase(){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }

    $this->load->model('temp_transaksi');
    $this->load->model('transaksi');

    $dataToTransfer = $this->temp_transaksi->getRawNoId();

```

```

foreach($dataToTransfer as $data){
    $this->transaksi->saveFromTemp($data);
}

$this->temp_transaksi->deleteAll();

echo "<script>
    alert('Berhasil menyimpan ke database');
    window.location='".site_url('dashboard')."';
</script>";
}

public function deleteAll(){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
            </script>";
        }

    if($this->session->
>userdata('prev_url') == site_url('bendahara/edit/editAwal')){
        $this->load->model('temp_transaksi', 'transaksi');
    }

    else if($this->session->
>userdata('prev_url') == site_url('bendahara/edit')){
        $this->load->model('transaksi');
    }
}

```



```

else{
    echo "<script>
        alert('Mohon akses dari halaman yang benar');
        window.location='".site_url('dashboard')."';
    </script>";
}

$this->transaksi->deleteAll();

redirect(site_url('bendahara/upload'));
}
}

```

5.1.1.5. Laporan

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Laporan extends CI_Controller {

    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }
        $this->load->model('transaksi');
    }
}

```

```

$donatur = $this->transaksi->getDonatur();
$periode=$this->transaksi->getPeriode();
$akun=$this->transaksi->getTotalAkun();
$t_periode=$this->transaksi->getPeriodeTahunan();
$counts = $this->transaksi->getAllSaldo();
$t_bulan = $this->transaksi->getPeriodeTahunBulan();
$kategori = $this->transaksi->getPeriodeTahunDetail();
$total = $this->transaksi->getTotal();
$data = array(
    'periode' => $periode,
    'donatur' => $donatur,
    'counts' => $counts,
    'akun' => $akun,
    't_periode' => $t_periode,
    'kategori' =>$kategori,
    'total' => $total,
    't_bulan' => $t_bulan,
);
$this->template-
>load("dashboard/template", "laporan/laporan", "Laporan", $data);
    //load("dashboard/template", "viewFolder/view", "Header")
}
}

```

5.1.1.6. Perencanaan

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class Perencanaan extends CI_Controller {
    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }
        $this->session->set_userdata('prev_url', current_url());

        $this->load->model('perencanaan_model');
        $allPerencanaan = $this->perencanaan_model->getAll();
        $data = array(
            'allPerencanaan' => $allPerencanaan
        );

        $this->template-
>load('dashboard/template', 'perencanaan/index', 'Perencanaan Penyaluran',
    $data);
    }

    public function addNew(){
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }
    }
}

```

```

    }
    else{
        $this->load->model('perencanaan_model');
        $data = $this->input->post(null, TRUE);

    }
}

public function calculator(){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('perencanaan_model');
        $this->load->model('rekap_model');

        $activePeriode = $this->rekap_model->getActivePeriode();
        if($activePeriode){
            $saldoSekarang = $this->perencanaan_model-
>getSaldo($activePeriode->id);
            $data = array(
                'saldoSekarang' => $saldoSekarang[0]->saldo_berjalan
            );
        }
    }
    else{

```

```

        $data = array(
            'saldoSekarang' => 0
        );
    }

    $this->template->
>load('dashboard/template', 'perencanaan/calculator', 'Kalkulator Perencan
aan', $data);
    }
}

public function create(){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('perencanaan_model');
        $data = $this->input->post(null, TRUE);

        if(isset($data['submit'])){
            $param = array(
                'untuk_periode' => $data['untuk_periode'],
                'jumlah_penerima' => $data['jumlah'],

```

```

        'nominal_penyalaran' => $data['nominal']
    );

    $this->perencanaan_model->create($param);

    redirect($this->session->userdata('prev_url'));
}

}

}

public function save(){
    if($this->session->userdata('role') != "bendahara"){
        echo "<script>
            alert('Maaf anda bukan bendahara');
            window.location='".site_url('dashboard')."';
        </script>";
    }
    else{
        $this->load->model('perencanaan_model');
        $data = $this->input->post(null, TRUE);

        if(isset($data['submit'])){
            $this->perencanaan_model->save($data);

            redirect($this->session->userdata('prev_url'));
        }
    }
}

```

```

    }

    public function delete(){
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }
        else{
            $this->load->model('perencanaan_model');
            $data = $this->input->post(null, TRUE);

            if(isset($data['submit'])){
                $this->perencanaan_model->delete($data['id']);

                redirect($this->session->userdata('prev_url'));
            }
        }
    }

    public function deleteStatus(){
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }
    }

```

```

else{
    $this->load->model('perencanaan_model');
    $data = $this->input->post(null, TRUE);

    if(isset($data['submit'])){
        $this->perencanaan_model->deleteStatus($data['status']);

        redirect($this->session->userdata('prev_url'));
    }
}
}
}
}

```

5.1.1.7. Rekapitulasi

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Rekap extends CI_Controller {

    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        $this->load->model('rekap_model', 'rekap');
    }
}

```



```

        $activePeriode = $this->rekap->getActivePeriode();
        if($activePeriode){
            $periodeBeforeActive = $this->rekap-
>getAllPeriodeBeforeActive($activePeriode->id);
            $periodikPertahun = $this->rekap-
>getPeriodikPertahun($activePeriode->id);
            $kumulatifPertahun = $this->rekap-
>getKumulatifPertahun($activePeriode->id);
            $periodikActiveTahun = $this->rekap-
>getActivePeriodikTahun($activePeriode->id);
            $kumulatifActiveTahun = $this->rekap-
>getActiveKumulatifTahun($activePeriode->id);
            $periodikPeriode = $this->rekap-
>getPeriodikPeriode($activePeriode->id);
            $kumulatifPeriode = $this->rekap-
>getKumulatifPeriode($activePeriode->id);
            $keteranganPeriode = $this->rekap-
>getKeteranganPeriodik($activePeriode->id);

        $data = array(
            'dataFetch' => TRUE,
            'activePeriode' => $activePeriode,
            'periodeBeforeActive' => $periodeBeforeActive,
            'periodikPertahun' => $periodikPertahun,
            'kumulatifPertahun' => $kumulatifPertahun,
            'periodikActiveTahun' => $periodikActiveTahun,
            'kumulatifActiveTahun' => $kumulatifActiveTahun,

```

```

        'periodikPeriode' => $periodikPeriode,
        'kumulatifPeriode' => $kumulatifPeriode,
        'keteranganPeriode' => $keteranganPeriode,
        'periodikCount' => count($periodikPeriode) - 1
    );
}
else{
    $data = array(
        'dataFetch' => FALSE
    );
}

$this->template->load('dashboard/template', 'rekap', 'Rekapitulasi', $data);
}
}

```

5.1.1.8. Surat perintah

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Suratperintah extends CI_Controller {

    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
            </script>";
        }
    }
}

```

```

        window.location='".site_url('dashboard')."';
    </script>";
}
$this->load->model('penerima');
$datas = $this->penerima->getAll();
$periode = $this->penerima->getPeriode();
$data = array(
    'datas' => $datas,
    'periode'=>$periode,
);
$this->template-
>load("dashboard/template", "suratperintah/suratPerintah", "Print Surat Pe
rintah", $data);
//load("dashboard/template", "viewFolder/view", "Header")
}

function __construct() {
    parent::__construct();
    $this->load->library('pdf');
}

function downloadLaporan(){
    $data = $this->input->post(null, TRUE);
    $dummy=array
    (
        array('Nama'=>"Nama 1", 'NRP'=>05111, 'norek'=>'no rek1'),
        array('Nama'=>"Nama 2", 'NRP'=>05114, 'norek'=>'no rek2'),
        array('Nama'=>"Nama 3", 'NRP'=>05113, 'norek'=>'no rek3')
    )
}

```

```

    );
    $this->load->model('penerima');
    $datas = $this->penerima->getNameByPeriode($data['periode']);

    $pdf = new FPDF('P', 'pt', 'A4');
    $pdf->AddPage();
    $pdf->SetFont('Arial', '', 14);
    $pdf->Cell(100, 16, "Surat Perintah Transfer",0,1);
    $pdf->Cell(100, 16, "Periode ".$data['periode'].",",0,1);
    $pdf->Cell(10,7,'',0,1);
    $pdf->SetFont('Arial', 'B',10);
    //$pdf->Cell(80,14,'NRP',1,0);
    $pdf->Cell(150,14,'Nama',1,0);
    $pdf->Cell(150,14,'Nomor Rekening',1,1);
    $pdf->SetFont('Arial', '',10);
    foreach ($datas as $row){
        //$pdf->Cell(80,14,$row->NRP,1,0);
        $pdf->Cell(150,14,$row->nama,1,0);
        $pdf->Cell(150,14,$row->norek,1,1);
    }
    $pdf->Output();
}
}

```

5.1.1.9. Upload

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class Upload extends CI_Controller {

    public function index()
    {
        if($this->session->userdata('role') != "bendahara"){
            echo "<script>
                alert('Maaf anda bukan bendahara');
                window.location='".site_url('dashboard')."';
            </script>";
        }

        $this->load->model('temp_transaksi');

        $data = $this->temp_transaksi->getAll();

        $count = 0;

        foreach($data as $d){
            $count++;
        }

        if($count > 0){
            echo "<script>
                alert('anda memiliki data yang belum disimpan');
                window.location='".site_url('bendahara/edit/editAwal')."';
            </script>";
        }
        else{

```

```

        $this->template-
>load("dashboard/template", "upload/index", "Upload File Mutasi");
    }

}

public function uploadAction(){
    $config = array(
        "upload_path" => FCPATH . "public/uploads/",
        "allowed_types" => 'csv'
    );

    $this->load->library('upload', $config);
    $this->upload->initialize($config);

    $field = 'userfile'; // The name attribute of the file input contr
ol.

    $file_selected = isset($_FILES[$field]) && isset($_FILES[$field]['
name']) && $_FILES[$field]['name'] != '';

    if($file_selected){
        if ( $this->upload->do_upload("userfile"))
        {
            $this->load->library('csvreader');

            $csv = $this->csvreader-
>parse_csv(FCPATH . 'public/uploads/UploadMutasi.csv');

```

```

$template = array('No', 'Tanggal', 'Deskripsi', 'Debit', 'Kredit', 'Periode');

$keys = $csv['keys'];
$datas = $csv['data'];

$count = 1;
$check = 0;

for ($i=1; $i < count($keys); $i++) {
    if($template[$i] != $keys[$i]){
        $check = 1;
    }
}

if(count($keys) != 6){
    $check = 1;
}

if($check == 0){
    foreach($datas as $data){
        if($data['Tanggal'] != ""){
            $check = 0;
            break;
        }
        else{
            $check = 2;
        }
    }
}

```

```

    }

    if($check == 2){
        echo "<script>
            alert('isi kosong');
            window.location='".site_url('bendahara/upload'
        )."'";

        "</script>";
    }
    else{
        $counter = 1;
        foreach($datas as $data){
            if($data['Tanggal'] == ""){
                unset($datas[$counter]);
            }
            $counter++;
        }

        $this->load->model('temp_transaksi');

        foreach($datas as $data){
            $this->temp_transaksi->create($data);
        }
    }
}
else{
    echo "<script>
        alert('format tidak sama');
    ";
}

```



```

        window.location='".site_url('bendahara/upload')."'
;

        </script>";
    }

    unlink(FCPATH . 'public/uploads/UploadMutasi.csv');

    echo "<script>
        alert('upload berhasil');
        window.location='".site_url('bendahara/edit/editAwal')
.'"';

        </script>";
    }
    else{
        $error = array('error' => $this->upload-
>display_errors());
        echo "<script>
            alert('upload gagal');
            window.location='".site_url('bendahara/upload/')."'";
            </script>";
        }
    }
    else{
        echo "<script>
            alert('file kosong');
            window.location='".site_url('bendahara/upload/')."'";

```

```

        </script>";
    }
}

public function downloadFormat(){
    $fileLocation = base_url("public/files/UploadMutasi.csv");
    $file = file_get_contents($fileLocation);

    force_download("UploadMutasi.csv", $file);
}
}

```

5.1.2. Models

5.1.2.1. Akun

```

<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Akun extends CI_Model
{
    private $_table = "akun";

    public $id;
    public $nama;

    public function getAll()
    {
        return $this->db->get($this->_table)->result();
    }
}

```

```

public function getById($id)
{
    return $this->db->get_where($this->_table, ["id" => $id])->row();
}

public function create($nama)
{
    $sql = "INSERT INTO akun (nama)" .
        " SELECT * FROM (SELECT ?) AS tmp" .
        " WHERE NOT EXISTS (" .
            "SELECT nama FROM akun WHERE nama = ?" .
            ") LIMIT 1;";

    $this->db->query($sql, [
        $nama,
        $nama
    ]);
}

public function deleteAll(){
    $this->db->delete($this->_table);
}

public function delete($id)
{
    return $this->db->delete($this->_table, array("id" => $id));
}
}

```

5.1.2.2. Donatur

```
<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Donatur extends CI_Model
{
    private $_table = "donatur";

    public $inisial;
    public $nama;

    public function getAll()
    {
        return $this->db->get($this->_table)->result();
    }

    public function save($data)
    {
        $this->inisial = $data['inisial'];
        $this->nama = $data['nama'];

        return $this->db->update($this->_table, $this,array('inisial' => $data['inisial']));
    }

    public function deleteAll(){
        $this->db->empty_table($this->_table);
    }
}
```

```

    public function delete($inisial)
    {
        return $this->db->delete($this->_table, array("inisial" => $inisial));
    }
}

```

5.1.2.3. Kategori

```

<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Kategori extends CI_Model
{
    private $_table = "kategori";

    public $id;
    public $nama;

    public function getAll()
    {
        return $this->db->get($this->_table)->result();
    }

    public function getById($id)
    {
        return $this->db->get_where($this->_table, ["id" => $id])->row();
    }
}

```

```

public function create($nama)
{
    $sql = "INSERT INTO kategori (nama)" .
        " SELECT * FROM (SELECT ?) AS tmp" .
        " WHERE NOT EXISTS (" .
            "SELECT nama FROM kategori WHERE nama = ?" .
            ") LIMIT 1;";

    $this->db->query($sql, [
        $nama,
        $nama
    ]);
}

public function deleteAll(){
    $this->db->delete($this->_table);
}

public function delete($id)
{
    return $this->db->delete($this->_table, array("id" => $id));
}
}

```

5.1.2.4. Penerima

```

<?php defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class Penerima extends CI_Model
{
private $_table = "beasiswa";

public function getAll()
{
return $this->db->query('select distinct
concat(DATE_FORMAT(beasiswa.tanggal_mulai,"%e/%m/%Y"),DATE_FORMAT(beasiswa
.tanggal_selesai," - %e/%m/%Y")) as periode_range,
biodata.no_rekening as norek,
biodata.nama_lengkap as nama
from beasiswa
right join pendaftar on pendaftar.beasiswa_id = beasiswa.beasiswa_id
left join biodata on biodata.biodata_id = pendaftar.biodata_id
left join calon on pendaftar.pendaftar_id = calon.pendaftar_id
left join penerima on penerima.calon_id = calon.calon_id')->result();
}

public function getPeriode()
{
return $this->db->query('select distinct
concat(DATE_FORMAT(beasiswa.tanggal_mulai,"%e/%m/%Y"),DATE_FORMAT(beasiswa
.tanggal_selesai," - %e/%m/%Y")) as periode_range
from beasiswa')->result();
}

public function getNameByPeriode($periode)
{

```

```

return $this->db->query('select distinct
concat(DATE_FORMAT(beasiswa.tanggal_mulai,"%e/%m/%Y"),DATE_FORMAT(beasiswa
.tanggal_selesai," - %e/%m/%Y")) as periode_range,
biodata.no_rekening as norek,
biodata.nama_lengkap as nama
from beasiswa
right join pendaftar on pendaftar.beasiswa_id = beasiswa.beasiswa_id
left join biodata on biodata.biodata_id = pendaftar.biodata_id
left join calon on pendaftar.pendaftar_id = calon.pendaftar_id
left join penerima on penerima.calon_id = calon.calon_id
where concat(DATE_FORMAT(beasiswa.tanggal_mulai,"%e/%m/%Y"),DATE_FORMAT(be
asiswa.tanggal_selesai," - %e/%m/%Y"))="' . $periode . "' )->result();
}
}

```

5.1.2.5. Perencanaan model

```

<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Perencanaan_model extends CI_Model
{
private $_table = "perencanaan";

public $id;
public $status = 'rencana';
public $untuk_periode;
public $nominal_penyaluran;
public $jumlah_penerima;

```



```
public function getAll()
{
    return $this->db->get($this->_table)->result();
}

public function getId($id)
{
    return $this->db->get_where($this->_table, ["id" => $id])->row();
}

public function create($data)
{
    $sql = "INSERT INTO perencanaan (untuk_periode, nominal_penyaluran, jumlah
    _penerima)
    VALUES (?, ?, ?);";

    $this->db->query($sql, [
        $data['untuk_periode'],
        $data['nominal_penyaluran'],
        $data['jumlah_penerima']
    ]);
}

public function save($data){
    $this->id = $data['id'];
    $this->status = $data['status'];
    $this->untuk_periode = $data['untuk_periode'];
    $this->nominal_penyaluran = $data['nominal_penyaluran'];
```

```

$this->jumlah_penerima = $data['jumlah_penerima'];

$this->db->update($this->_table, $this, array('id' => $this->id));
}

public function getSaldo($activePeriode){
$sql = 'SELECT sum(saldo) as saldo_berjalan FROM `transaksi` WHERE periode
<=?;';

return $this->db->query($sql,[$activePeriode])->result();
}

public function deleteStatus($status){
$this->db->delete($this->_table, array('status' => $status));
}

public function deleteAll(){
$this->db->empty_table($this->_table);
}

public function delete($id)
{
return $this->db->delete($this->_table, array("id" => $id));
}
}

```

5.1.2.6. Periode

```

<?php defined('BASEPATH') OR exit('No direct script access allowed');

```

```
class Periode extends CI_Model
{
    private $_table = "periode";

    public $id;
    public $nama;
    public $deskripsi;
    public $status;

    public function getAll()
    {
        return $this->db->get($this->_table)->result();
    }

    public function save($data)
    {
        $this->id = $data['id'];
        $this->nama = $data['nama'];
        $this->deskripsi = $data['deskripsi'];
        $this->status = $data['status'];

        return $this->db->update($this->_table, $this,array('id' => $data['id']));
    }

    public function deleteAll(){
        $this->db->empty_table($this->_table);
    }
}
```

```

    }

    public function delete($id)
    {
        return $this->db->delete($this->_table, array("id" => $id));
    }
}

```

5.1.2.7. Rekap model

```

<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Rekap_model extends CI_Model
{
    public function getActivePeriode(){
        $sql = "select id from periode where status = 1 limit 1;";

        return $this->db->query($sql)->row();
    }

    public function getAllPeriodeBeforeActive($active){
        $sql = "select id from periode where substring(id,1,4)=substring(?
,1,4) and id < ? order by id asc;";

        return $this->db->query($sql, [$active, $active])->result();
    }

    public function getPeriodikPertahun($activePeriode){

```

```

        $sql = "select
            substring(periode,1,4) as periode_tahun,
            IFNULL((select count(y.id) from transaksi y where substring(y.peri
ode,1,4)=substring(x.periode,1,4) and y.id_jenistransaksi='1' and y.id_kat
egori='1' and y.id_akun='2'), 0) as jumlah_donasi,
            IFNULL((select sum(y.saldo) from transaksi y where substring(y.per
iode,1,4)=substring(x.periode,1,4) and y.id_jenistransaksi='1' and y.id_ka
tegori='1' and y.id_akun='2'), 0) as nominal_donasi,
            IFNULL((select count(y.id) from transaksi y where substring(y.peri
ode,1,4)=substring(x.periode,1,4) and y.id_jenistransaksi='2' and y.id_kat
egori='4' and y.id_akun='2'), 0) as jumlah_penerima,
            IFNULL((select sum(y.saldo) from transaksi y where substring(y.per
iode,1,4)=substring(x.periode,1,4) and y.id_jenistransaksi='2' and y.id_ka
tegori='4' and y.id_akun='2'), 0) as nominal_penyialuran,
            IFNULL((select sum(y.saldo) from transaksi y where substring(y.per
iode,1,4)=substring(x.periode,1,4) and y.id_kategori='3'), 0) as lain_lain
        from transaksi x
        where substring(x.periode,1,4)<substring(?,1,4)
        group by substring(periode, 1, 4);";

        return $this->db->query($sql,[$activePeriode])>result();
    }

    public function getKumulatifPertahun($activePeriode){
        $sql = "select
            substring(periode,1,4) as periode_tahun,

```

```

        IFNULL((select count(y.id) from transaksi y where substring(y.periode,1,4)<=substring(x.periode,1,4) and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2'), 0) as jumlah_donasi,
        IFNULL((select sum(y.saldo) from transaksi y where substring(y.periode,1,4)<=substring(x.periode,1,4) and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2'), 0) as nominal_donasi,
        IFNULL((select count(y.id) from transaksi y where substring(y.periode,1,4)<=substring(x.periode,1,4) and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'), 0) as jumlah_penerima,
        IFNULL((select sum(y.saldo) from transaksi y where substring(y.periode,1,4)<=substring(x.periode,1,4) and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'), 0) as nominal_penyaluran,
        IFNULL((select sum(y.saldo) from transaksi y where substring(y.periode,1,4)<=substring(x.periode,1,4) and y.id_kategori='3'), 0) as lain_lain
    from transaksi x
    where substring(x.periode,1,4)<substring(?,1,4)
    group by substring(periode, 1, 4);

    return $this->db->query($sql,[$activePeriode])->result();
}

public function getActiveKumulatifTahun($activePeriode){
    $sql = "select
        periode,
        IFNULL((select count(y.id) from transaksi y where y.periode<=x.periode and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2'),
        0) as jumlah_donasi,

```

```

        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.periode and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2'), 0) as nominal_donasi,

        IFNULL((select count(y.id) from transaksi y where y.periode<=x.periode and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'), 0) as jumlah_penerima,

        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.periode and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'), 0) as nominal_penyaluran,

        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.periode and y.id_kategori='3'), 0) as lain_lain

        from transaksi x
        where x.periode=?
        group by periode;";

    return $this->db->query($sql,[$activePeriode])>result();
}

public function getActivePeriodikTahun($activePeriode){
    $sql = "select
        periode,
        substring(periode,1,4) as periode_pertahun,
        IFNULL((select count(y.id) from transaksi y where y.periode<=x.periode and substring(y.periode,1,4)=substring(x.periode,1,4) and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2'), 0) as jumlah_donasi
    ,
        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.periode and substring(y.periode,1,4)=substring(x.periode,1,4) and y.id_jenis

```

```

transaksi='1' and y.id_kategori='1' and y.id_akun='2'), 0) as nominal_donasi,

        IFNULL((select count(y.id) from transaksi y where y.periode<=x.periode and substring(y.periode,1,4)=substring(x.periode,1,4) and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'), 0) as jumlah_penerima,

        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.periode and substring(y.periode,1,4)=substring(x.periode,1,4) and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'), 0) as nominal_penyaluran,

        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.periode and substring(y.periode,1,4)=substring(x.periode,1,4) and y.id_kategori='3'), 0) as lain_lain

        from transaksi x

        where x.periode<=?

        group by periode

        order by periode DESC

        limit 1;";

    return $this->db->query($sql,[$activePeriode])->result();
}

public function getPeriodikPeriode($activePeriode){
    $sql = "select

        periode, periode.nama as nama_periode,

        IFNULL((select count(y.id) from transaksi y where y.periode=x.periode and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2'),

        0) as jumlah_donasi,

```



```

        IFNULL((select sum(y.saldo) from transaksi y where y.periode=x.periode and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2'),
        0) as nominal_donasi,
        IFNULL((select count(y.id) from transaksi y where y.periode=x.periode and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'),
        0) as jumlah_penerima,
        IFNULL((select sum(y.saldo) from transaksi y where y.periode=x.periode and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'),
        0) as nominal_penyaluran,
        IFNULL((select sum(y.saldo) from transaksi y where y.periode=x.periode and y.id_kategori='3'), 0) as lain_lain
    from transaksi x
    left join periode on periode.id = x.periode
    where substring(x.periode,1,4)=substring(?,1,4) and x.periode<=?
    group by periode;";

    return $this->db->query($sql,[$activePeriode, $activePeriode])->result();
}

public function getKumulatifPeriode($activePeriode){
    $sql = "select
    periode,
    IFNULL((select count(y.id) from transaksi y where y.periode<=x.periode and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2'),
    0) as jumlah_donasi,

```

```

        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.pe
riode and y.id_jenistransaksi='1' and y.id_kategori='1' and y.id_akun='2')
, 0) as nominal_donasi,
        IFNULL((select count(y.id) from transaksi y where y.periode<=x.per
iode and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2'),
0) as jumlah_penerima,
        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.pe
riode and y.id_jenistransaksi='2' and y.id_kategori='4' and y.id_akun='2')
, 0) as nominal_penyaluran,
        IFNULL((select sum(y.saldo) from transaksi y where y.periode<=x.pe
riode and y.id_kategori='3'), 0) as lain_lain
    from transaksi x
    where substring(x.periode,1,4)=substring(?,1,4) and x.periode<=?
    group by periode;";

    return $this->db->query($sql,[$activePeriode, $activePeriode])-
>result();
}

public function getKeteranganPeriodik($activePeriode){
    $sql = "select id,
        IFNULL((select deskripsi from periode y where y.id=x.id), '-
') as deskripsi
    from periode x
    where substring(x.id,1,4)=substring(?,1,4) and x.id<=?";

    return $this->db->query($sql,[$activePeriode, $activePeriode])-
>result();

```

```
}
}
```

5.1.2.8. Temporary Transaksi

```
<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Temp_transaksi extends CI_Model
{
    private $_table = "temp_transaksi";

    public $id;
    public $deskripsi;
    public $debit;
    public $kredit;
    public $saldo;
    public $periode;
    public $tanggal;
    public $id_jenistransaksi;
    public $id_kategori;
    public $id_akun;

    public function getAll()
    {
        $this->db->select('temp_transaksi.*, jenis_transaksi.nama as jenis_transaksi, kategori.nama as kategori, akun.nama as akun');
        $this->db->from($this->_table);
```

```

        $this->db-
>join('jenis_transaksi', 'jenis_transaksi.id = id_jenistransaksi', 'inner'
);

        $this->db->join('kategori', 'kategori.id = id_kategori', 'inner');
        $this->db->join('akun', 'akun.id = id_akun', 'inner');
        $this->db->order_by('periode', 'DESC');
        $this->db->order_by('akun', 'ASC');
        $this->db->order_by('tanggal', 'DESC');
        $query = $this->db->get();
        return $query->result();
    }

    public function getRawNoId(){
        $this->db-
>select('deskripsi, debit, kredit, saldo, periode, tanggal, id_jenistransa
ksi, id_kategori, id_akun, inisial_donatur');
        $this->db->from($this->_table);
        $query = $this->db->get();
        return $query->result();
    }

    public function getById($id)
    {
        return $this->db->get_where($this->_table, ["id" => $id])->row();
    }

    public function getAkun(){
        $this->db->select('*');
    }

```

```
$this->db->from('akun');  
$query = $this->db->get();  
return $query->result();  
}  
  
public function getJenisTrans(){  
    $this->db->select('*');  
    $this->db->from('jenis_transaksi');  
    $query = $this->db->get();  
    return $query->result();  
}  
  
public function getKategori(){  
    $this->db->select('*');  
    $this->db->from('kategori');  
    $query = $this->db->get();  
    return $query->result();  
}  
  
public function create($data)  
{  
    $this->deskripsi = $data['Deskripsi'];  
    $this->debit = $data['Debit'];  
    $this->kredit = $data['Kredit'];  
    $this->saldo = $this->kredit - $this->debit;  
    $this->periode = $data['Periode'];  
    $this->tanggal = date("Y-m-d", strtotime($data['Tanggal']));
```

```

        if($this->saldo >=0){
            $this->id_jenistransaksi = 1;
        }
        else{
            $this->id_jenistransaksi = 2;
        }
        $this->id_kategori = 3;
        $this->id_akun = 1;

        $this->db->insert($this->_table, $this);
    }

    public function save($data){
        $this->id = $data['id'];
        $this->deskripsi = $data['deskripsi'];
        $this->debit = $data['debit'];
        $this->kredit = $data['kredit'];
        $this->saldo = $this->kredit - $this->debit;
        $this->periode = $data['periode'];
        $this->tanggal = date("Y-m-d", strtotime($data['tanggal']));
        if($this->saldo >=0){
            $this->id_jenistransaksi = 1;
        }
        else{
            $this->id_jenistransaksi = 2;
        }
        $this->id_kategori = $data['kategori'];
        $this->id_akun = $data['akun'];
    }

```

```

        $this->inisial_donatur = $data['inisial_donatur'];

        return $this->db->update($this->_table, $this,array('id' => $data['id']));
    }

    public function split($data){
        $this->deskripsi = $data['deskripsi'];
        $this->debit = $data['debit'];
        $this->kredit = $data['kredit'];
        $this->id_kategori = $data['kategori'];
        $this->id_akun = $data['akun'];
        $this->periode = $data['periode'];
        $this->tanggal = date("Y-m-d", strtotime($data['tanggal']));
        $this->inisial_donatur = $data['inisial_donatur'];

        $this->saldo = $this->kredit - $this->debit;
        if($this->saldo >=0){
            $this->id_jenistransaksi = 1;
        }
        else{
            $this->id_jenistransaksi = 2;
        }

        $this->db->insert($this->_table, $this);
    }

    public function createBalance($data){

```

```

$this->deskripsi = 'Penyeimbang akun untuk transfer';
$this->debit = $data['kredit'];
$this->kredit = $data['debit'];
$this->id_kategori = $data['kategori'];
$this->id_akun = $data['akun'];
$this->periode = $data['periode'];
$this->tanggal = date("Y-m-d", strtotime($data['tanggal']));
$this->inisial_donatur = $data['inisial_donatur'];

$this->saldo = $this->kredit - $this->debit;
if($this->saldo >=0){
    $this->id_jenistransaksi = 1;
}
else{
    $this->id_jenistransaksi = 2;
}

$this->db->insert($this->_table, $this);
}

public function transfer($data){
    $this->deskripsi = $data['deskripsi'];
    $this->debit = $data['debit'];
    $this->kredit = $data['kredit'];
    $this->id_kategori = $data['kategori'];
    $this->id_akun = $data['akun_tujuan'];
    $this->periode = $data['periode'];
    $this->tanggal = date("Y-m-d", strtotime($data['tanggal']));

```



```

        $this->inisial_donatur = $data['inisial_donatur'];

        $this->saldo = $this->kredit - $this->debit;
        if($this->saldo >=0){
            $this->id_jenistransaksi = 1;
        }
        else{
            $this->id_jenistransaksi = 2;
        }

        $this->db->insert($this->_table, $this);
    }

    public function deleteAll(){
        $this->db->empty_table($this->_table);
    }

    public function delete($id)
    {
        return $this->db->delete($this->_table, array("id" => $id));
    }
}

```

5.1.2.9. Transaksi

```

<?php defined('BASEPATH') OR exit('No direct script access allowe
d');

class Transaksi extends CI_Model

```

```

{
private $_table = "transaksi";

public $id;
public $deskripsi;
public $debit;
public $kredit;
public $saldo;
public $periode;
public $tanggal;
public $id_jenistransaksi;
public $id_kategori;
public $id_akun;
public $inisial_donatur;

public function getAll()
{
$this->db-
>select('transaksi.*, jenis_transaksi.nama as jenis_transaksi, ka
tegori.nama as kategori, akun.nama as akun');
$this->db->from($this->_table);
$this->db-
>join('jenis_transaksi', 'jenis_transaksi.id = id_jenistransaksi'
, 'inner');
$this->db-
>join('kategori', 'kategori.id = id_kategori', 'inner');
$this->db->join('akun', 'akun.id = id_akun', 'inner');
$this->db->order_by('periode', 'DESC');

```

```

$this->db->order_by('akun', 'ASC');
$this->db->order_by('tanggal', 'DESC');
$query = $this->db->get();
return $query->result();
}

public function search($data){
$sql = "SELECT transaksi.* , jenis_transaksi.nama as jenis_transaksi, kategori.nama as kategori, akun.nama as akun
FROM transaksi
inner join jenis_transaksi
on jenis_transaksi.id = id_jenistransaksi
inner join kategori
on kategori.id = id_kategori
inner join akun
on akun.id = id_akun
WHERE deskripsi LIKE ? AND debit LIKE ?
AND kredit LIKE ? AND saldo LIKE ?
AND periode LIKE ? AND tanggal LIKE ?
AND id_jenistransaksi LIKE ? AND id_kategori LIKE ?
AND id_akun LIKE ? AND inisial_donatur LIKE ?;";

return $this->db->query($sql,[
$data['deskripsi_s'],
$data['debit_s'],
$data['kredit_s'],
$data['saldo_s'],
$data['periode_s'],

```

```

$data['tanggal_s'],
$data['id_jenis_s'],
$data['id_kategori_s'],
$data['id_akun_s'],
$data['donatur_s']
])->result();
}

public function saveFromTemp($data){
$this->deskripsi = $data->deskripsi;
$this->debit = $data->debit;
$this->kredit = $data->kredit;
$this->saldo = $data->saldo;
$this->periode = $data->periode;
$this->tanggal = date("Y-m-d", strtotime($data->tanggal));
$this->id_jenistransaksi = $data->id_jenistransaksi;
$this->id_kategori = $data->id_kategori;
$this->id_akun = $data->id_akun;
$this->inisial_donatur = $data->inisial_donatur;

$this->updateDonatur($this->inisial_donatur);
$this->updatePeriode($this->periode);

$this->db->insert($this->_table, $this);
}

public function getById($id)
{

```

```
return $this->db->get_where($this->_table, ["id" => $id])->row();
}

public function getAkun(){
    $this->db->select('*');
    $this->db->from('akun');
    $query = $this->db->get();
    return $query->result();
}

public function getJenisTrans(){
    $this->db->select('*');
    $this->db->from('jenis_transaksi');
    $query = $this->db->get();
    return $query->result();
}

public function getKategori(){
    $this->db->select('*');
    $this->db->from('kategori');
    $query = $this->db->get();
    return $query->result();
}

public function create($data)
{
    $this->deskripsi = $data['Deskripsi'];
    $this->debit = $data['Debit'];
```

```

$this->kredit = $data['Kredit'];
$this->saldo = $this->kredit - $this->debit;
$this->periode = $data['Periode'];
$this->tanggal = date("Y-m-d", strtotime($data['Tanggal']));

if($this->saldo >=0){
    $this->id_jenistransaksi = 1;
}
else{
    $this->id_jenistransaksi = 2;
}
$this->id_kategori = 3;
$this->id_akun = 1;

$this->db->insert($this->_table, $this);
}

public function save($data){
    $this->id = $data['id'];
    $this->deskripsi = $data['deskripsi'];
    $this->debit = $data['debit'];
    $this->kredit = $data['kredit'];
    $this->saldo = $this->kredit - $this->debit;
    $this->periode = $data['periode'];
    $this->tanggal = date("Y-m-d", strtotime($data['tanggal']));
    if($this->saldo >=0){
        $this->id_jenistransaksi = 1;
    }
}

```

```

else{
$this->id_jenistransaksi = 2;
}
$this->id_kategori = $data['kategori'];
$this->id_akun = $data['akun'];
$this->inisial_donatur = $data['inisial_donatur'];

$this->updateDonatur($this->inisial_donatur);
$this->updatePeriode($this->periode);

return $this->db->update($this->_table, $this,array('id' => $data['id']));
}

public function split($data){
$this->deskripsi = $data['deskripsi'];
$this->debit = $data['debit'];
$this->kredit = $data['kredit'];
$this->id_kategori = $data['kategori'];
$this->id_akun = $data['akun'];
$this->periode = $data['periode'];
$this->tanggal = date("Y-m-d", strtotime($data['tanggal']));
$this->inisial_donatur = $data['inisial_donatur'];

$this->saldo = $this->kredit - $this->debit;
if($this->saldo >=0){
$this->id_jenistransaksi = 1;
}
}

```

```
else{
$this->id_jenistransaksi = 2;
}

$this->db->insert($this->_table, $this);
}

public function createBalance($data){
$this->deskripsi = 'Penyeimbang akun untuk transfer';
$this->debit = $data['kredit'];
$this->kredit = $data['debit'];
$this->id_kategori = $data['kategori'];
$this->id_akun = $data['akun'];
$this->periode = $data['periode'];
$this->tanggal = date("Y-m-d", strtotime($data['tanggal']));
$this->inisial_donatur = $data['inisial_donatur'];

$this->updateDonatur($this->inisial_donatur);
$this->updatePeriode($this->periode);

$this->saldo = $this->kredit - $this->debit;
if($this->saldo >=0){
$this->id_jenistransaksi = 1;
}
else{
$this->id_jenistransaksi = 2;
}
```



```

$this->db->insert($this->_table, $this);
}

public function transfer($data){
$this->deskripsi = $data['deskripsi'];
$this->debit = $data['debit'];
$this->kredit = $data['kredit'];
$this->id_kategori = $data['kategori'];
$this->id_akun = $data['akun_tujuan'];
$this->periode = $data['periode'];
$this->tanggal = date("Y-m-d", strtotime($data['tanggal']));
$this->inisial_donatur = $data['inisial_donatur'];

$this->updateDonatur($this->inisial_donatur);
$this->updatePeriode($this->periode);

$this->saldo = $this->kredit - $this->debit;
if($this->saldo >=0){
$this->id_jenistransaksi = 1;
}
else{
$this->id_jenistransaksi = 2;
}

$this->db->insert($this->_table, $this);
}

public function deleteAll(){

```

```

$this->db->empty_table($this->_table);
}

public function delete($id)
{
return $this->db->delete($this->_table, array("id" => $id));
}

public function getPeriode(){
return $this->db-
>query('select distinct periode as id, x.nama as nama, (select y.
nama from periode y where y.id=x.id-
1) as namasebelum, x.deskripsi as deskripsi from transaksi left j
oin periode x ON transaksi.periode=x.id')->result();
}

public function getDonatur(){
return $this->db
-
>query('select inisial_donatur as donatur, periode from transaksi
where id_kategori=1 and saldo>0')
->result();
}

public function getByPeriode($data)
{

```

```

$this->db-
>select('transaksi.*, jenis_transaksi.nama as jenis_transaksi, ka
tegori.nama as kategori, akun.nama as akun');
$this->db->from($this->_table);
$this->db-
>join('jenis_transaksi', 'jenis_transaksi.id = id_jenistransaksi'
, 'inner');
$this->db-
>join('kategori', 'kategori.id = id_kategori', 'inner');
$this->db->join('akun', 'akun.id = id_akun', 'inner');
$query = $this->db->get();
return $query->result();
}

public function getAllSaldo()
{
return $this->db->query("Select distinct
periode,
IFNULL((select sum(saldo) from transaksi b where b.periode=a.peri
ode and b.id_jenistransaksi=1),0) as penerimaan,
IFNULL((select sum(saldo) from transaksi b where b.periode=a.peri
ode and b.id_jenistransaksi=2),0) as pengeluaran,
IFNULL((select sum(saldo) from transaksi b where b.periode<a.peri
ode),0) as periodeSebelum,
IFNULL((select sum(saldo) from transaksi b where b.periode<=a.per
iode),0) as saldo
from transaksi a")->result();

```

```

}
public function getTotalAkun()
{
return $this->db->query("Select
akun.nama as nama,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and id_jenistransaksi='1'),0) as penerimaan_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and id_jenistransaksi='2'),0) as pengeluaran_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun),0) as saldo_tahun
from transaksi x LEFT JOIN akun ON x.id_akun = akun.id
GROUP BY id_akun")->result();
}
public function getPeriodeTahunan()
{
return $this->db->query("select
akun.nama as nama,
substring(periode,1,4) as periode_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and substring(y.periode,1,4)=substring(x.periode,1,4) and i
d_jenistransaksi='1'),0) as penerimaan_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and substring(y.periode,1,4)=substring(x.periode,1,4) and i
d_jenistransaksi='2'),0) as pengeluaran_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and substring(y.periode,1,4)=substring(x.periode,1,4)),0) a
s saldo_tahun

```

```

from transaksi x LEFT JOIN akun
ON x.id_akun = akun.id
group by id_akun, substring(periode,1,4)"->result();
}

public function getPeriodeTahunBulan()
{
return $this->db->query("select distinct
akun.nama as nama,
substring(periode,1,4) as periode_tahun,
periode.nama as nama_bulan,
periode,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and x.periode=y.periode and id_jenistransaksi='1'),0) as pe
nerimaan_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and x.periode=y.periode and id_jenistransaksi='2'),0) as pe
ngeluaran_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and x.periode=y.periode),0) as saldo_tahun
from transaksi x left join akun on x.id_akun=akun.id left join pe
riode on periode.id=x.periode)"->result();
}

public function getPeriodeTahunDetail()
{
return $this->db->query("select distinct
akun.nama as nama,
kategori.nama as kategori,
substring(periode,1,4) as periode_tahun,

```

```

periode,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and x.periode=y.periode and id_jenistransaksi='1' and x.id_
kategori=y.id_kategori),0) as penerimaan_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and x.periode=y.periode and id_jenistransaksi='2' and x.id_
kategori=y.id_kategori),0) as pengeluaran_tahun,
IFNULL((select sum(y.saldo) from transaksi y where x.id_akun=y.id
_akun and x.periode=y.periode and x.id_kategori=y.id_kategori),0)
as saldo_tahun
from transaksi x left join akun on x.id_akun=akun.id left join ka
tegori on x.id_kategori=kategori.id")->result();
}
public function getTotal()
{
return $this->db->query("Select
IFNULL((select sum(saldo) from transaksi where id_jenistransaksi=
'1'),0) as penerimaan_total,
IFNULL((select sum(saldo) from transaksi where id_jenistransaksi=
'2'),0) as pengeluaran_total,
IFNULL(sum(saldo),0) as saldo_total
from transaksi")->result();
}
/*
public function getPeriodeTahunanAkumulasi()
{
return $this->db->query("select
akun.nama,

```

```

substring(periode,1,4) as periode_tahun,
IFNULL((select sum(y.saldo) from transaksi y where substring(y.pe
riode,1,4)<=substring(x.periode,1,4) and id_jenistransaksi='1'),0
) as penerimaan_tahun,
IFNULL((select sum(y.saldo) from transaksi y where substring(y.pe
riode,1,4)<=substring(x.periode,1,4) and id_jenistransaksi='2'),0
) as pengeluaran_tahun,
IFNULL((select sum(y.saldo) from transaksi y where substring(y.pe
riode,1,4)<=substring(x.periode,1,4)),0) as saldo_tahun
from transaksi x LEFT JOIN akun
ON x.id_akun = akun.id
group by id_akun, substring(periode,1,4)"->result();
}
*/
public function updateDonatur($donatur){
if($donatur != ""){
$sql = "INSERT INTO donatur (inisial)" .
" SELECT * FROM (SELECT ?) AS tmp" .
" WHERE NOT EXISTS (" .
"SELECT inisial FROM donatur WHERE inisial = ?" .
") LIMIT 1;";

$this->db->query($sql, [
$donatur,
$donatur
]);
}
}
}

```

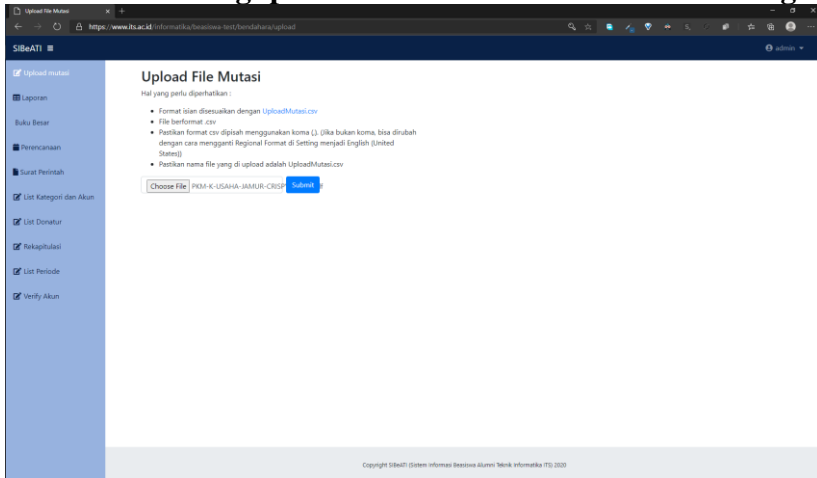
```
public function updatePeriode($periode){
    if($periode != ""){
        $sql = "INSERT INTO periode (id)" .
        " SELECT * FROM (SELECT ?) AS tmp" .
        " WHERE NOT EXISTS (" .
        "SELECT id FROM periode WHERE id = ?" .
        ") LIMIT 1;";

        $this->db->query($sql, [
            $periode,
            $periode
        ]);
    }
}
```


5.2. Implementasi Antarmuka Pengguna

Gambar 5.2 Antarmuka UC002

5.2.3. Mengupload dokumen berisi mutasi rekening



Gambar 5.3 Antarmuka UC003

5.2.4. Mengedit data transaksi yang tercatat

Edit

Tanggal: 2020-09-06
Format: yyyy-mm-dd (contoh: 2020-12-31)

Deskripsi: Penyetorangan akun untuk transfer

Debit: 0
Format: tanpa titik, dengan koma, tidak kosong (contoh: 2000000)

Kredit: 1500000
Format: tanpa titik, dengan koma, tidak kosong (contoh: 2000000)

Periode: 2020/9
Format: tahun-yyymm (contoh: 2020/12)

Save

No	Tanggal	Deskripsi	Debit	Kredit	Periode	Jenis	Akron
6	2020-09-06	Penyetorangan akun untuk transfer	0	1500000	2020/9	Penerimaan	Bank Lainnya

Logout Your screenshot is copied to clipboard

Gambar 5.4 Antarmuka UC004

5.2.5. Membuat perencanaan penyaluran beasiswa

Hitung

Hitung: 2

Hitung nominal yang bisa di salurkan per orang: 1100000

Untuk periode: Bulan-Tahun

Simpan

Hitung nominal penyaluran

Nominal Penyaluran: Hitung

No	Tanggal	Deskripsi	Debit	Kredit	Periode	Jenis	Akron
6	2020-09-06	Penyetorangan akun untuk transfer	0	1500000	2020/9	Penerimaan	Bank Lainnya

Copyright SibaATI Sistem Informasi Beasiswa Akademi Teknik Informatika ITS 2020

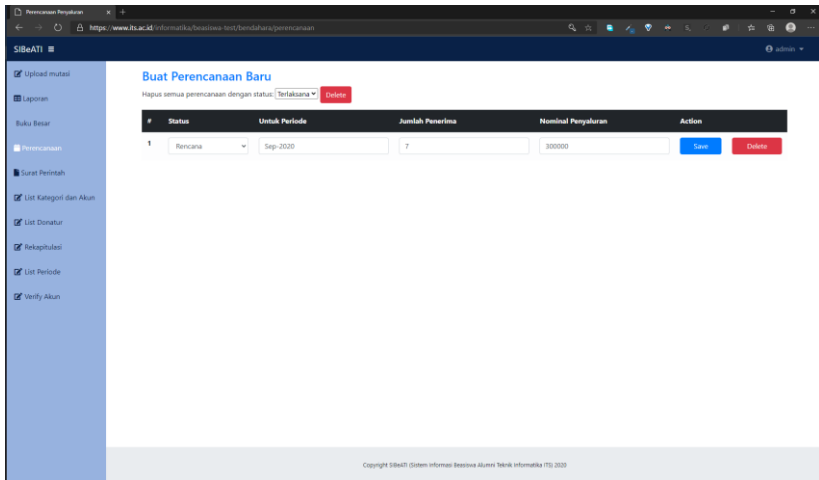
Gambar 5.5 Antarmuka UC005

5.2.6. Menghapus data transaksi yang tercatat

#	Tanggal	Deskripsi	Debit	Kredit	Saldo	Periode	Jenis Transaksi	Akun	Kategori	Alias Donatur	Action
1	2020-09-06	Penyeimbang akun untuk transfer	0	1500000	1500000	202010	Penerimaan	Bank	Lainnya		Edit Delete
2	2020-09-06	Transfer ke rekening Mandiri000	1500000	0	(1500000)	202010	Pengeluaran	Bank	Lainnya		Edit Delete
3	2020-09-06	Transfer ke rekening Mandiri000	1500000	0	(1500000)	202009	Pengeluaran	Bank	Lainnya		Edit Delete
4	2020-09-06	Transfer ke rekening Mandiri000	1500000	0	(1500000)	202009	Pengeluaran	Bank	Lainnya		Edit Delete
5	2020-09-06	Transfer ke rekening Mandiri000-123	1500000	0	(1500000)	202009	Pengeluaran	Bank	Lainnya		Edit Delete
6	2020-09-06	Transfer ke rekening BC0000	1500000	0	(1500000)	202009	Pengeluaran	Bank	Lainnya		Edit Delete

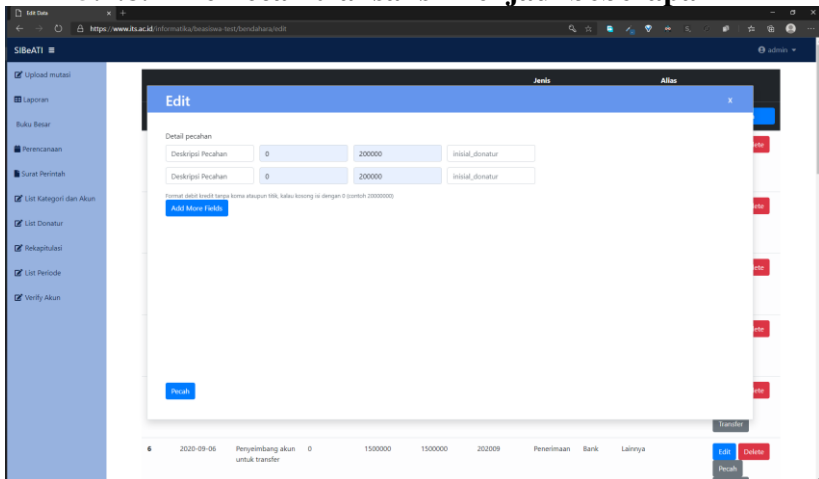
Gambar 5.6 Antarmuka UC006

5.2.7. Melihat perencanaan penyaluran beasiswa



Gambar 5.7 Antarmuka UC007

5.2.8. Memecah transaksi menjadi beberapa



Gambar 5.8 Antarmuka UC008

5.2.9. Melihat laporan kegiatan

View Laporan / Ringkasan / Periode: 202007

"Laporan Beasiswa Alumni T. Informatika"

"A. Penyerima & Pengeluaran Periode Juli-2020",

1. Donatur

2. Posisi Keuangan

Saldo Akhir Bulan: Rp.0

Penerimaan Bulan Juli-2020: Rp.6.900.000

Penyuluhan Bulan Juli-2020: Rp.31.000

Saldo Akhir Juli-2020: Rp.5.969.000

"B. Penyuluhan Beasiswa Alumni ITI,"

1.

2. Bagi para donatur beasiswa dapat diklaimkan langsung ke rekening berikut.

Bank Mandiri Cabang Kota Kediri

No Rekening 171-00-1010111-4

an Fadeli Sukya (T. Informatika ITS Angkatan 2000)

Terdapat Saldo Penerimaan penyuluhan Periode Juli-2020

Copyright SIBeATI Sistem Informasi Beasiswa Alumni Teknik Informatika ITS 2020

Gambar 5.9 Antarmuka UC009

5.2.10. Melihat laporan keuangan

Rekapitulasi Beasiswa Alumni T. Informatika - ITS

Periode	Periodik					Kumulatif					Saldo Berjalan	Keterangan	
	Jumlah Donasi	Nominal Donasi	Jumlah Penerima	Nominal Penyuluhan	Lain2	Jumlah Donasi	Nominal Donasi	Jumlah Penerima	Nominal Penyuluhan	Lain2			
2020	2	2.000.000	0	0	0	2	2.000.000	0	0	0	0	11.907.000	-
Juli-2020	0	0	0	0	0	0	0	0	0	0	0	5.969.000	
Aug-2020	0	0	0	0	0	0	0	0	0	0	0	14.938.000	
Sep-2020	2	2.000.000	0	0	0	2	2.000.000	0	0	0	0	9.907.000	

Penyuluhan pada bulan Juni-2020, Telah disalurkan beasiswa kepada 7 mahasiswa masing-masing Rp.1.000.000,- (untuk 3 bulan) * Nadhwa Annisa H (511194000013) * Muhammad Marqis Abdul Jall (511194000014) * Aul Syahya (5111940000106) * Albertus Tommy Helmi Putra (5111940000108) * Richard Asmarakandi (511194000017) * Putri Endah Pujiastari (511174000030) * Renu Agung Purnama (511194000013)

Copyright SIBeATI Sistem Informasi Beasiswa Alumni Teknik Informatika ITS 2020

5.2.12. Melakukan transfer antar akun

Transfer

Deskripsi:

Akun Tujuan:

Tanggal:
Format harus yyyy-mm-dd (contoh 2020-12-31)

Transfer

6 2020-09-06 Penyeimbang akun untuk transfer 0 150000 150000 202009 Penerimaan Bank Lainnya

Gambar 5.12 Antarmuka UC012

5.2.13. Membuat surat perintah penyaluran

Penerima Beasiswa

Berikut adalah Penerima beasiswa periode:

Nama	Nomor Rekening
Hur Muhammad Husnul Habib	1430018423341
Nafza	
Tukli Yensar Ratna	100200
Izzah Diniyalla	1400016587785

Surat Perintah Transfer:

Copyright Sibeati Sistem Informasi Beasiswa akurasi Nafis Informatika ITS 2020

Gambar 5.13 Antarmuka UC013

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap SIBeATI. Pengujian dilakukan untuk memastikan kualitas sistem informasi yang dibangun dan kesesuaian hasil eksekusi sistem informasi dengan analisis dan perancangan perangkat lunak.

6.1. Skenario Pengujian

6.1.1. Melakukan pencatatan transaksi donasi

- a. Melakukan login
- b. Menekan menu buku besar pada sidebar
- c. Menekan tombol edit pada transaksi yang diinginkan
 - i. Jenis Transaksi : Pemasukan
 - ii. Akun : Beasiswa
 - iii. Kategori : Donasi
 - iv. Inisial Donatur : <donatur>
- d. Menekan tombol save

6.1.2. Melakukan pencaatatan penyaluran beasiswa

- a. Melakukan login
- b. Menekan menu buku besar pada sidebar
- c. Menekan tombol edit pada transaksi yang diinginkan
 - i. Jenis Transaksi : Pengeluaran
 - ii. Akun : Beasiswa
 - iii. Kategori : Penyaluran
- d. Menekan tombol save

6.1.3. Mengupload dokumen berisi mutase rekening

- a. Melakukan login
- b. Masuk ke halaman upload rekening

- c. Menekan choose file
- d. Memilih file yang disediakan
- e. Menekan upload

6.1.4. Mengedit data transaksi yang tercatat

- a. Melakukan login
- b. Masuk ke halaman buku besar
- c. Menekan tombol edit pada transaksi
- d. Edit data sesuai keinginan
- e. Tekan tombol save

6.1.5. Membuat perencanaan penyaluran beasiswa

- a. Melakukan login
- b. Membuka menu perencanaan pada sidebar
- c. Menekan link Buat Perencanaan Baru
- d. Memilih ingin menghitung berdasar apa
- e. Mengisi data
- f. Menekan tombol hitung
- g. Mengisi data rencana untuk periode berapa
- h. Menekan tombol save

6.1.6. Menghapus data transaksi yang tercatat

- a. Melakukan login
- b. Membuka menu buku besar pada sidebar
- c. Menekan tombol hapus pada transaksi

6.1.7. Melihat perencanaan penyaluran beasiswa

- a. Melakukan login
- b. Membuka menu perencanaan pada sidebar

6.1.8. Memecah transaksi menjadi beberapa

- a. Melakukan login

- b. Membuka menu buku besar pada sidebar
- c. Menekan tombol pecah
- d. Memasukkan detail pemecahan (deskripsi dan nominal)
- e. Menekan tombol save

6.1.9. Melihat laporan kegiatan

- a. Melakukan login
- b. Menekan tombol laporan pada sidebar
- c. Memilih tipe laporan ringkasan pada drop down
- d. Memilih periode yang ingin di tampilkan

6.1.10. Melihat laporan keuangan

- a. Melakukan login
- b. Membuka menu rekapitulasi pada sidebar

6.1.11. Melihat neraca keuangan

- a. Melakukan login
- b. Menekan tombol laporan pada sidebar
- c. Memilih tipe laporan neraca pada drop down

6.1.12. Melakukan transfer antar akun

- a. Melakukan login
- b. Membuka menu buku besar pada sidebar
- c. Menekan tombol transfer pada transaksi
- d. Memilih ingin transfer ke akun mana
- e. Menekan tombol transfer

6.1.13. Membuat surat perintah penyaluran beasiswa

- a. Melakukan login
- b. Menekan menu surat perintah pada sidebar
- c. Memilih periode surat perintah yang ingin di buat

- d. Tekan print surat perintah untuk pergi ke halaman surat perintah untuk save atau langsung cetak

6.2. Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Hasil evaluasi pengujian dapat dilihat pada Tabel 6.1.

Tabel 6.1 : Tabel Evaluasi Pengujian Aplikasi Sesuai Kebutuhan

No.	Kebutuhan	Status
UC001	Melakukan pencatatan transaksi donasi	Berhasil
UC002	Melakuykan pencatatan penyaluran beasiswa	Berhasil
UC003	Mengupload dokumen berisi mutase rekening	Berhasil
UC004	Mengedit data transaksi yang tercatat	Berhasil
UC005	Membuat perencanaan penyaluran beasiswa	Berhasil
UC006	Menghapus data transaksi yang tercatat	Berhasil
UC007	Melihat perencanaan penyaluran beasiswa	Berhasil
UC008	Memecah transaksi menjadi beberapa transaksi	Berhasil
UC009	Melihat laporan kegiatan	Berhasil
UC010	Melihat laporan keuangan	Berhasil
UC011	Melihat neraca keuangan	Berhasil
UC012	Melakukan transfer antar akun	Berhasil
UC013	Membuat surat perintah penyaluran	Berhasil

BAB VII

KESIMPULAN DAN SARAN

Kesimpulan yang didapat setelah melakukan pembuatan SIBeATI adalah sebagai berikut:

- a. Sistem informasi yang dibangun oleh mahasiswa telah sesuai dengan permintaan dan dapat dioperasikan oleh pengguna baik oleh mahasiswa, voter, maupun tim seleksi.
- b. Dengan adanya SIBeATI, pengelolaan beasiswa alumni Teknik Informatika ITS lebih terintegrasi dan kualitas tata kelola beasiswa lebih meningkat.
- c. *Framework* yang digunakan dalam pembangunan SIBeATI ini merupakan *framework* yang umum digunakan di lingkungan kerja yaitu *framework* CodeIgniter sehingga mudah untuk *maintenance* sistem informasi tersebut.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Wikipedia (2020). Aplikasi web - Wikipedia bahasa Indonesia, ensiklopedia bebas. [online] Available at: https://id.wikipedia.org/wiki/Sistem_informasi [Accessed 14 Agustus 2019].
- [2] Harison, Ahmad Syarif. 2016. Sistem Informasi Geografis Sarana Pada Kabupaten Pasaman Barat. *Jurnal Teknologi dan Sistem Komputer*. 4(2):90-94.
- [3] Herman Yuliansyah. 2014. Perancangan Replikasi Basis Data MySQL Dengan Mekanisme Pengamanan Menggunakan SSL Encryption. *Jurnal Informatika*. 8(1):826-836.
- [4] Randi Proska Sandra, Ahmaddul Hadi, dan Yasdinul Huda. 2017. Perancangan Aplikasi Clickhost Berbasis Web Menggunakan Metode HMVC (Hierarchical Model View Controller) Sebagai Sistem Pencairan dan Promosi Indekos di Kota Padang. *Jurnal Vokasional Teknik Elektronika & Informatika*. 5(2).
- [5] Logique (2020). Apa itu Codeigniter dan Keunggulannya? | Logique [online] Available at: <https://www.logique.co.id/blog/2018/08/13/codeigniter-dan-keunggulannya/> [Accessed 14 Agustus 2020].
- [6] UMBY Repository (2020). Electronic Journal-UMBY Repository [online] Available at: <http://eprints.mercubuana-yogya.ac.id/2513/9/BAB%20II.pdf> [Accessed 14 Agustus 2019].
- [7] Yuki Yanuar Ratna, Izzah Dinillah. 2020. Spesifikasi Kebutuhan Perangkat Lunak untuk SIBeATI (Sistem Kepesertaan).

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Nur Muhammad Husnul Habib Yahya, lahir pada tanggal 03 Maret 1999. Penulis merupakan mahasiswa yang sedang menempuh studi di Departemen Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS).



Carlo Win Marshal, lahir pada 05 Juni 1999. Penulis merupakan mahasiswa yang sedang menempuh studi di Departemen Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS).